

CSD481	MINI PROJECT	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PWS	0	0	3	4	2019

Preamble: The objective of this course is to apply the fundamental concepts of different courses learned in respective Minor Streams: Software Engineering, Machine Learning and Networking. This course helps the learners to get an exposure to the development of application software/hardware solutions/ software simulations in the field of Computer Science and Engineering. It enables the learners to understand the different steps to be followed such as literature review and problem identification, preparation of requirement specification & design document, testing, development and deployment. Mini project enables the students to boost their skills, widen the horizon of thinking and their ability to resolve real life problems.

Prerequisite:

A sound knowledge in courses studied in respective minor stream.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)
CO2	Identify and survey the relevant literature for getting exposed to related solutions. (Cognitive Knowledge Level: Apply)
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern**Mark Distribution**

Total Marks	CIE Marks	ESE Marks
150	75	75

Continuous Internal Evaluation Pattern:

Attendance	10 marks
Project Guide	15 marks
Project Report	10 marks
Evaluation by the Committee (will be evaluating the level of completion and demonstration of functionality/specifications, presentation, oral examination, work knowledge and involvement)	: 40 marks

Student Groups with 4 or 5 members should identify a topic of interest in consultation with a Faculty Advisor/Project Coordinator/Guide. Review the literature and gather information pertaining to the chosen topic. State the objectives and develop a methodology to achieve the objectives. Carryout the design/fabrication or develop codes/programs to achieve the objectives by strictly following steps specified in the teaching plan. Innovative design concepts, performance, scalability, reliability considerations, aesthetics/ergonomic, user experience and security aspects taken care of in the project shall be given due weight.

The progress of the mini project is evaluated based on a minimum of two reviews. The review committee may be constituted by a senior faculty member, Mini Project coordinator and project guide. The internal evaluation shall be made based on the progress/outcome of the project, reports and a viva-voce examination, conducted internally by a 3-member committee. A project report is required at the end of the semester. The project has to be demonstrated for its full design specifications.

End Semester Examination Pattern:

The marks will be distributed as

Presentation	: 30 marks
Demo	: 20 marks
Viva	: 25 marks.
Total	: 75 marks.

TEACHING PLAN

Students are expected to follow the following steps.

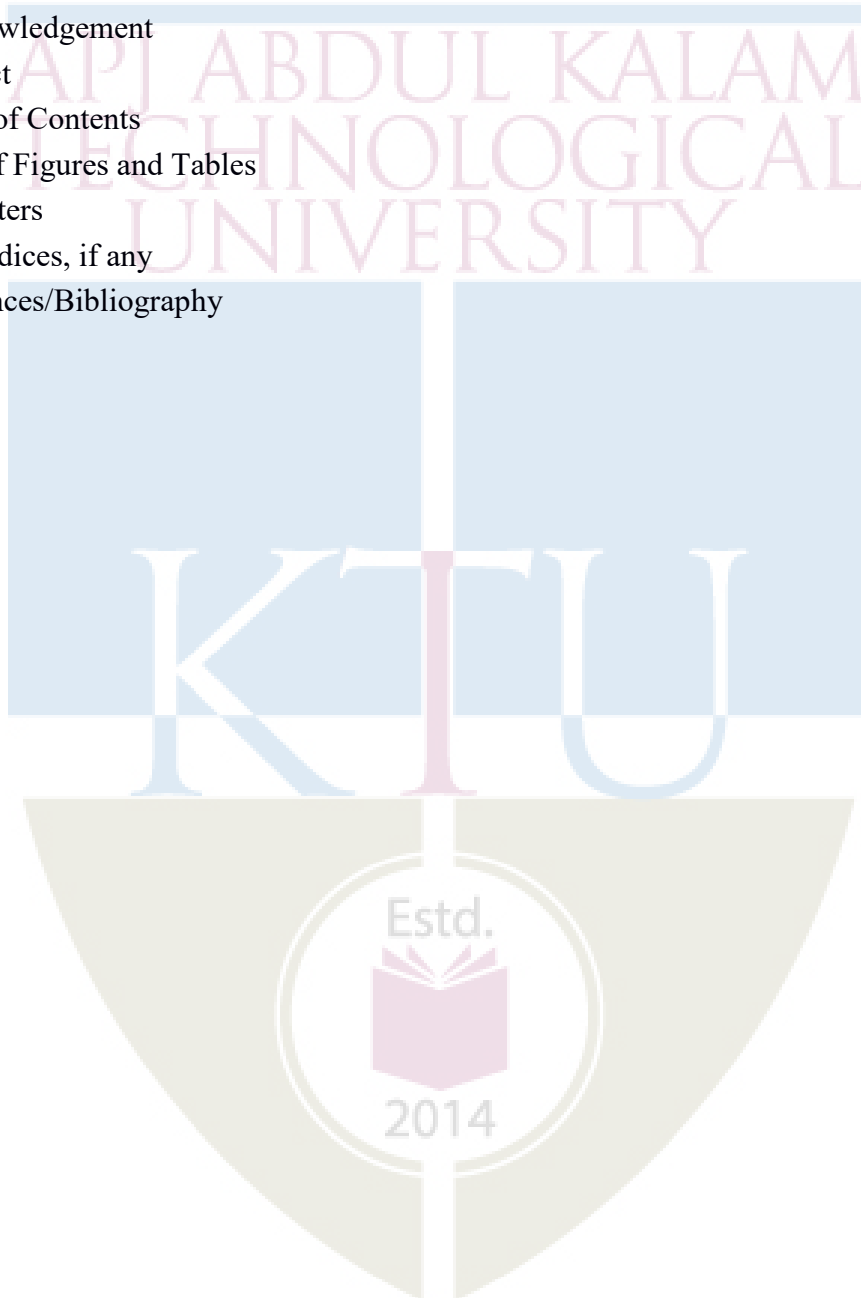
1. Review of Literature and Identification of a problem
2. Create an abstract with a problem statement, solution approach, technology stack, team, etc.
3. Create Requirements Specification
4. Create Design Document . This may include designs like,
 - a. System Architecture Design
 - b. Application Architecture Design
 - c. GUI Design
 - d. API Design
 - e. Database Design
 - f. Technology Stack
5. Deployment, Test Run & Get Results
6. Prepare Project Report

Guidelines for the Report preparation

A bonafide report on the mini project shall be submitted within one week after the final presentation. Minimum number of pages should be 40.

- Use Times New Roman font for the entire report – Chapter/Section Title – Times New Roman 18, Bold; Heading 2 – Times New Roman 16, Bold; Heading 3 – Times New Roman 14, Bold; Body- Times New Roman 12, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.
- Alignments – Chapter/Section Title – Center, Heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.

- Figures & Tables – Ensure that all Figures and Tables are suitably numbered and given proper names/headings. Write figure title under the figure and table title above the table.
- **Suggestive order of documentation:**
 - i. Top Cover
 - ii. Title page
 - iii. Certification page
 - iv. Acknowledgement
 - v. Abstract
 - vi. Table of Contents
 - vii. List of Figures and Tables
 - viii. Chapters
 - ix. Appendices, if any
 - x. References/Bibliography



CSD482	MINI PROJECT	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PWS	0	0	3	4	2019

Preamble: The objective of this course is to apply the fundamental concepts of different courses learned in respective Minor Streams: Software Engineering, Machine Learning and Networking. This course helps the learners to get an exposure to the development of application software/hardware solutions/ software simulations in the field of Computer Science and Engineering. It enables the learners to understand the different steps to be followed such as literature review and problem identification, preparation of requirement specification & design document, testing, development and deployment. Mini project enables the students to boost their skills, widen the horizon of thinking and their ability to resolve real life problems.

Prerequisite: A sound knowledge in courses studied in respective minor stream.

Course Outcomes: After the completion of the course the student will be able to

CO#	CO
CO1	Identify technically and economically feasible problems (Cognitive Knowledge Level: Apply)
CO2	Identify and survey the relevant literature for getting exposed to related solutions. (Cognitive Knowledge Level: Apply)
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques (Cognitive Knowledge Level: Apply)
CO4	Prepare technical report and deliver presentation (Cognitive Knowledge Level: Apply)
CO5	Apply engineering and management principles to achieve the goal of the project (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												
CO3												
CO4												
CO5												

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern**Mark Distribution**

Total Marks	CIE Marks	ESE Marks
150	75	75

Continuous Internal Evaluation Pattern:

Attendance	10 marks
Project Guide	15 marks
Project Report	10 marks
Evaluation by the Committee (will be evaluating the level of completion and demonstration of functionality/specifications, presentation, oral examination, work knowledge and involvement)	: 40 marks

Student Groups with 4 or 5 members should identify a topic of interest in consultation with a Faculty Advisor/Project Coordinator/Guide. Review the literature and gather information pertaining to the chosen topic. State the objectives and develop a methodology to achieve the objectives. Carryout the design/fabrication or develop codes/programs to achieve the objectives by strictly following steps specified in the teaching plan. Innovative design concepts, performance, scalability, reliability considerations, aesthetics/ergonomic, user experience and security aspects taken care of in the project shall be given due weight.

The progress of the mini project is evaluated based on a minimum of two reviews. The review committee may be constituted by a senior faculty member, Mini Project coordinator and project guide. The internal evaluation shall be made based on the progress/outcome of the project, reports and a viva-voce examination, conducted internally by a 3-member committee. A project report is required at the end of the semester. The project has to be demonstrated for its full design specifications.

End Semester Examination Pattern:

The marks will be distributed as

Presentation	: 30 marks
Demo	: 20 marks
Viva	: 25 marks.
Total	: 75 marks.

TEACHING PLAN

Students are expected to follow the following steps.

1. Review of Literature and Identification of a problem
2. Create an abstract with a problem statement, solution approach, technology stack, team, etc.
3. Create Requirements Specification
4. Create Design Document . This may include designs like,
 - a. System Architecture Design
 - b. Application Architecture Design
 - c. GUI Design
 - d. API Design
 - e. Database Design
 - f. Technology Stack
5. Deployment, Test Run & Get Results
6. Prepare Project Report

Guidelines for the Report preparation

A bonafide report on the mini project shall be submitted within one week after the final presentation. Minimum number of pages should be 40.

- Use Times New Roman font for the entire report – Chapter/Section Title – Times New Roman 18, Bold; Heading 2 – Times New Roman 16, Bold; Heading 3 – Times New Roman 14, Bold; Body- Times New Roman 12, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.
- Alignments – Chapter/Section Title – Center, Heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.
- Figures & Tables – Ensure that all Figures and Tables are suitably numbered and given proper names/headings. Write figure title under the figure and table title above the table.

- **Suggestive order of documentation:**

- i. Top Cover
- ii. Title page
- iii. Certification page
- iv. Acknowledgement
- v. Abstract
- vi. Table of Contents
- vii. List of Figures and Tables
- viii. Chapters
- ix. Appendices, if any
- x. References/Bibliography



CST 281	OBJECT ORIENTED PROGRAMMING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		MINOR	3	1	0	4	2019

Preamble: This is the programming course for awarding B.Tech. Minor in Computer Science and Engineering with specialization in **Software Engineering**. The purpose of this course is to enable learners to solve problems by breaking it down to object level while designing software and to implement it using Java. This course covers Object Oriented Principles, Object Oriented Programming in Java, Inheritance, Exception handling, Event handling, multithreaded programming and working with window-based graphics. This course helps the learners to develop Mobile applications, Enterprise Applications, Scientific Applications and Web based Applications.

Prerequisite: Topics covered under the course PROGRAMMING IN C (EST 102)

Course Outcomes: After the completion of the course the student will be able to

CO1	Write Java programs using the object oriented concepts - classes, objects, constructors, data hiding, inheritance and polymorphism (Cognitive Knowledge Level: Apply)
CO2	Utilise datatypes, operators, control statements, built in packages & interfaces, Input/ Output Streams and Files in Java to develop programs (Cognitive Knowledge Level: Apply)
CO3	Illustrate how robust programs can be written in Java using exception handling mechanism (Cognitive Knowledge Level: Understand)
CO4	Write application programs in Java using multithreading (Cognitive Knowledge Level: Apply)
CO5	Write Graphical User Interface based application programs by utilising event handling features and Swing in Java (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcome

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓						✓		✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓								✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks (%)
	Test1 (Marks %)	Test2 (Marks %)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

SYLLABUS

Object Oriented Programming Using Java

Module 1

Introduction:

Approaches to Software Design - Functional Oriented Design, Object Oriented Design, Case Study of Automated Fire Alarm System.

Object Modeling Using UML – Basic Object Oriented concepts, UML (Unified Modeling Language) diagrams, Use case model, Class diagram, Interaction diagram, Activity diagram, State chart diagram.

Introduction to Java - Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. Java Virtual Machine (JVM), Java compiler, Bytecode, Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues.

Module 2

Core Java Fundamentals:

Primitive Data types - Integers, Floating Point Types, Characters, Boolean. Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class.

Operators - Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence.

Control Statements - Selection Statements, Iteration Statements and Jump Statements.

Object Oriented Programming in Java - Class Fundamentals, Declaring Objects, Object Reference, Introduction to Methods, Constructors, *this* Keyword, Method Overloading, Using Objects as Parameters, Returning Objects, Recursion, Access Control, Static Members, Final Variables, Inner Classes, Command-Line Arguments, Variable Length Arguments.

Module 3

More features of Java:

Inheritance - Super Class, Sub Class, The Keyword super, protected Members, Calling Order of Constructors, Method Overriding, the Object class, Abstract Classes and Methods, Using final with Inheritance.

Packages and Interfaces - Defining Package, CLASSPATH, Access Protection, Importing Packages, Interfaces.

Exception Handling - Checked Exceptions, Unchecked Exceptions, *try* Block and *catch* Clause, Multiple *catch* Clauses, Nested *try* Statements, *throw*, *throws* and *finally*.

Module 4

Advanced features of Java:

Input/Output - I/O Basics, Reading Console Input, Writing Console Output, PrintWriter Class, Object Streams and Serialization, Reading and Writing Files.

Java Library - String Handling – String Constructors, String Length, Special String Operations - Character Extraction, String Comparison, Searching Strings, Modifying Strings, Using valueOf(), Comparison of StringBuffer and String.

Collections framework – Collections overview, Collections Class – ArrayList. Accessing Collections via an Iterator.

Module 5

GUI Programming, Event Handling and Multithreaded Programming:

Swing fundamentals - Swing Key Features, Model View Controller (MVC), Swing Controls, Components and Containers, Exploring Swing - JFrame, JLabel, JButton, JTextField.

Event handling - Event Handling Mechanisms, Delegation Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Using the Delegation Model.

Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread, Creating Multiple Threads, Suspending, Resuming and Stopping Threads.

Text Books:

1. Herbert Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011.
2. Rajib Mall, Fundamentals of Software Engineering, 4th edition, PHI, 2014.
3. Paul Deitel, Harvey Deitel, Java How to Program, Early Objects 11th Edition, Pearson, 2018.

Reference Books:

1. Y. Daniel Liang, Introduction to Java Programming, 7/e, Pearson, 2013.
2. Nageswararao R., Core Java: An Integrated Approach, Dreamtech Press, 2008.
3. Flanagan D., Java in A Nutshell, 5/e, O'Reilly, 2005.
4. Barclay K., J. Savage, Object Oriented Design with UML and Java, Elsevier, 2004.
5. Sierra K., Head First Java, 2/e, O'Reilly, 2005.
6. Balagurusamy E., Programming JAVA a Primer, 5/e, McGraw Hill, 2014.

Sample Course Level Assessment Questions

Course Outcome1(CO1): For the following passage develop UML diagrams and then implement it as a Java program in accordance with your UML design.

Passage: College Office collects semester fee and college bus fee for each student. A clerk at the college office collects the fees from each student. The bus fee is calculated depending on the distance of the corresponding bus stop from the college. The semester fee varies depending upon the semester as well as branch of each student. Students are supposed to pay the fees in full. Economically backward students are eligible for 50% discount in semester fee. The consolidated fees receipt is issued to each student by the clerk, which contains the student name, admission number, semester and branch of student along with details of fees collected. Students can log in and view the details of fees remitted and dues if any. The system allows students and clerk level login to the system. Clerk is able to view reports of each class showing status of fees payment of each student.

Course Outcome 2 (CO2): Write a Java program to prepare the rank list of students based on their performance in the first Semester B.Tech. Degree examination at APJ Abdul Kalam Technological University. The output should be stored in a file.

Course Outcome 3 (CO3): Write a program to demonstrate how event handling and exception handling are supported in Java..

Course Outcome 4 (CO4): Write a program to demonstrate the start, run, sleep and join methods in Thread class..

Model Question Paper

QP CODE:

PAGES:3

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

THIRD SEMESTER B.TECH (MINOR) DEGREE EXAMINATION, MONTH & YEAR

Course Code: CST 281

Course Name: Object Oriented Programming using Java

Max.Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Briefly explain why Java is considered to be secure and portable.
2. Describe the concept of association among classes with an example.
3. Explain the different arithmetic operators in Java.
4. Explain the use for command line arguments with a suitable Java program
5. Explain the use of CLASSPATH with an example.
6. What are the different types of exceptions?
7. Explain file handling features available in Java.
8. Write a simple program to read and print an integer value in Java.
9. Explain the concept of *main thread* in multi-threading.
10. Explain any two Event classes in Java.

Part B

Answer any one question completely from each module

- 11.
- (a) Describe in detail polymorphism, abstraction and inheritance with suitable examples. (9)
 - (b) What is Java Virtual Machine? (5)

OR

- 12.
- (a) Compare and contrast Functional Oriented and Object Oriented approach by considering a simple bus ticket reservation system. (5)
 - (b) What is a class diagram? Explain with an example. (9)

- 13.
- (a) Explain primitive data types in Java. How are they different from other data types? (8)
 - (b) Explain variables and arrays in Java. (6)

OR

- 14.s
- (a) Using a suitable Java program explain the concept of methods and constructors. (8)
 - (b) Explain the keyword **super** and its usage in Java. (6)

- 15.
- (a) Using a table, explain the effect of access specifiers in inheritance. (6)
 - (b) Describe in detail about exception handling using **try** block and **catch** clause in Java with the help of a suitable Java program. (8)

OR

- 16.
- (a) What is an interface in Java? Explain with a suitable example. (8)
 - (b) Explain **throw**, **throws** and **finally** constructs with the help of a Java program. (6)

17.

- (a) Explain *ArrayList* collections framework. Also explain the use of iterator in accessing collections. (8)
- (b) Bring out difference between “==” and *equals()* method with the help of a sample program (6)

OR

18.

- (a) Compare Byte Streams and Character Streams. Write a program to demonstrate the usage of the *PrintWriter* class. (8)
- (b) Explain any three String constructors with the help of sample code for each. (6)

19.

- (a) Explain in detail the Delegation Event model for event handling in Java. (7)
- (b) Describe in detail the creation of a thread using the Runnable interface. (7)

OR

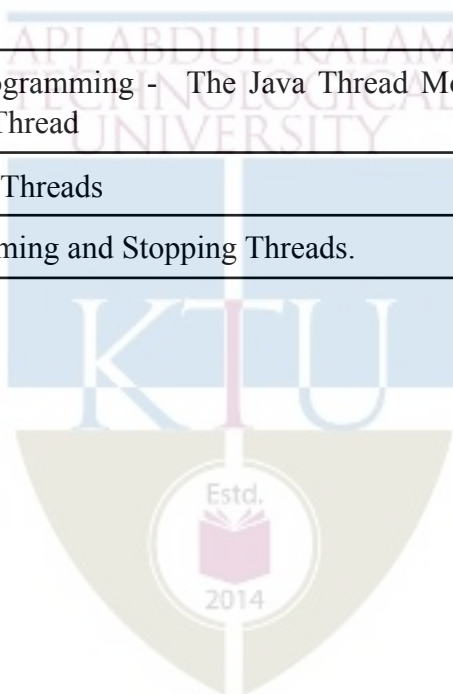
20.

- (a) What are the differences between a process and a thread? (4)
- (b) Write a Graphical User Interface (GUI) based Java program to implement a simple calculator supporting the operations addition, subtraction, multiplication and division. Use Swing controls to implement GUI. There may be three text boxes, the first two for operands and the last for result. Add four buttons for the above operations. Write neat comments in your program to show how you handle events. (10)

Teaching Plan		
Module 1 (Introduction)		(8 hours)
1.1	Approaches to Software Design- Functional Oriented Design, Object-Oriented Design, Case Study of Automated Fire Alarm System.	1 hour
1.2	Object Modeling Using UML – Basic object oriented concepts	1 hour
1.3	Basic object oriented concepts	1 hour
1.4	UML diagrams, Use case model	1hour
1.5	Class diagram, Interaction diagram	1hour
1.6	Activity diagram, State chart diagram	1hour
1.7	Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. JVM, Java compiler, Bytecode	1hour
1.8	Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues	1hour
Module 2 (Core Java Fundamentals)		(12 hours)
2.1	Primitive Data types - Integers, Floating Point Types, Characters, Boolean	1 hour
2.2	Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class.	1 hour
2.3	Operators - Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence.	1 hour
2.4	Control Statements - Selection Statements, Iteration Statements and Jump Statements.	1 hour
2.5	Object Oriented Programming in Java - Class Fundamentals, Declaring Objects	1 hour
2.6	Object Reference, Introduction to Methods	1 hour
2.7	Constructors, <i>this</i> Keyword	1 hour
2.8	Method Overloading, Using Objects as Parameters	1 hour

2.9	Returning Objects, Recursion	1 hour
2.10	Access Control, static Members	1 hour
2.11	Final Variables, Inner Classes	1 hour
2.12	Command-Line Arguments, Variable Length Arguments	1 hour
Module 3 (More features of Java)		(8 hours)
3.1	Inheritance - Super class, Sub class, the keyword super, protected Members,	1 hour
3.2	Calling Order of Constructors, Method Overriding, the Object class,	1 hour
3.3	Abstract Classes and Methods, Using final with Inheritance	1 hour
3.4	Packages and Interfaces - Defining Package, CLASSPATH, Access Protection, Importing Packages	1 hour
3.5	Interfaces	1 hour
3.6	Exception Handling - Checked Exceptions, Unchecked Exceptions, <i>try</i> Block and <i>catch</i> Clause	1 hour
3.7	Multiple <i>catch</i> Clauses, Nested <i>try</i> Statements	1 hour
3.8	<i>throw</i> , <i>throws</i> and <i>finally</i>	1 hour
Module 4 (Advanced features of Java)		(8 hours)
4.1	Input/Output - I/O Basics, Reading Console Input	1hour
4.2	Writing Console Output, PrintWriter Class	1hour
4.3	Object Streams and Serialization	1hour
4.4	Serialization, Working with Files	1hour
4.5	Working with Files	1hour
4.6	Java Library - String Handling – String Constructors, String Length, Special String Operations	1hour
4.7	Character Extraction, String Comparison, Searching Strings, Modifying Strings Using <code>valueOf()</code> , Comparison of <code>StringBuffer</code> and <code>String</code> .	1hour
4.8	Collections framework – Collections overview, Collections Class – <code>ArrayList</code> . Accessing Collections via an Iterator.	1hour

Module 5 (GUI Programming, Event Handling and Multithreaded Programming)		(9 hours)
5.1	Swings fundamentals, Swing Key Features	
5.2	MVC, Swing Controls, Components and Containers	
5.3	Exploring Swing –JFrame, JLabel, JButton, JTextField.	
5.4	Event handling - Event Handling Mechanisms, Delegation Event Model	1hour
5.5	Delegation Event Model, Event Classes	1hour
5.6	Sources of Events, Event Listener Interfaces, Using the Delegation Model	1hour
5.7	Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread	1hour
5.8	Creating Multiple Threads	1hour
5.9	Suspending, Resuming and Stopping Threads.	1hour



CST 282	Programming Methodologies	Category	L	T	P	CREDIT	YEAR OF INTRODUCTION
		MINOR	3	1	0	4	2019

Preamble: This is the second course for awarding B.Tech Minor in Computer Science and Engineering with specialization in *Software Engineering*. The course provides the learners a clear understanding of the main constructs of contemporary programming languages and the various systems of ideas that have been used to guide the design of programming languages. This course covers the concepts of Names, Bindings & Scope, Statement-Level Control Structures, Sub Programs, support for Object Oriented Programming, Exception Handling, Event Handling, Concurrency Control, Functional Programming and Logic Programming. This course helps the learners to equip with the knowledge necessary for the critical evaluation of existing and upcoming programming languages. It also enables the learner to choose the most appropriate language for a given programming task, apply that language's approach to structure or organize the code and classify programming languages based on their features.

Prerequisite:

1. Topics covered under the course Programming in C (EST 102)
2. Object Oriented Programming (CST 251)

Course Outcomes: After the completion of the course the student will be able to

CO1	Explain the criteria for evaluating programming languages and compare Imperative, Functional and Logic programming languages (Cognitive Knowledge Level: Understand)
CO2	Explain the characteristics of data types and variables (Cognitive Knowledge Level: Understand)
CO3	Illustrate how control flow structures and subprograms help in developing the structure of a program to solve a computational problem (Cognitive Knowledge Level: Apply)
CO4	Explain the characteristics of Object Oriented Programming Languages (Cognitive Knowledge Level: Understand)
CO5	Compare concurrency constructs in different programming languages (Cognitive Knowledge Level: Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓							✓		✓
CO2	✓	✓								✓		✓
CO3	✓	✓	✓	✓								✓
CO4	✓	✓								✓		✓
CO5	✓	✓	✓									✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination (Marks%)
	Test 1 (Marks%)	Test 2 (Marks%)	
Remember	30	30	30
Understand	50	50	50
Apply	20	20	20
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Test : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

SYLLABUS

Module 1

Introduction – Role of Programming Languages, Programming Domains, Language Evaluation Criteria, Influence on Language Design, Language Design Trade-offs, Implementation Methods. **Names, Bindings & Scope** – Names, Variables, Concept of Binding, Scope and Lifetime, Referencing Environments.

Module 2

Data Types – Primitive Data Types, Character String Types, User-Defined Ordinal Types, Array Types, Record Types, List Types, Pointer & Reference Types, Type Checking, Strong Typing, Type Equivalence. **Expressions** – Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation. **Assignment** - Assignment Statements, Mixed-mode Assignment.

Module 3

Statement-Level Control Structures – Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands. **Subprograms** – Design Issues of Subprograms, Local Referencing Environments, Parameter Passing Methods, Subprograms as Parameters, Overloaded Subprograms, Closures, Co-routines.

Module 4

Support for Object Oriented Programming – Inheritance, Dynamic Binding, Design Issues for Object Oriented Languages, Support for Object Oriented Programming in C++, Implementation of Object-Oriented Constructs. **Exception Handling** – Basic Concepts, Design Issues. **Event Handling** - Introduction to Event Handling.

Module 5

Concurrency – Subprogram Level Concurrency, Semaphores, Monitors, Message Passing. **Functional Programming Languages** – Introduction to LISP and Scheme, Comparison of Functional and Imperative Languages. **Logic Programming Languages** – Basic Elements of Prolog, Applications of Logic Programming.

Text Books

1. Robert W. Sebesta, Concepts of Programming Languages, 10th Edition, Pearson.
2. Scott M. L., Programming Language Pragmatics, 3rd Edn., Morgan Kaufmann Publishers.

Reference Books:

1. Kenneth C. Loudon, Programming Languages: Principles and Practice, 2nd Edn., Cengage Learning.
2. Tucker A. B. and R. E. Noonan, Programming Languages: Principles and Paradigms, 2nd Edn. –TMH.
3. Ravi Sethi, Programming Languages: Concepts & Constructs, 2nd Edn., Pearson Education.
4. David A. Watt, Programming Language Design Concepts, Wiley Dreamtech

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): Compare any three programming languages based on the language evaluation criteria. Prepare a list of characteristics that affect the language evaluation criteria. Identify the advantages and disadvantages of imperative, functional and logic programming languages.

Course Outcome 2 (CO2): Two most important design issues that are specific to character string types are (1) whether a string is simply a special kind of character array or a primitive type (2) whether strings have static or dynamic length. Justify your answer.

Course Outcome 3 (CO3):

1. Describe three situations where a combined counting and logical looping statement is needed.
2. Describe the ways that aliases can occur with pass-by-reference parameters.
3. Identify the two fundamental design considerations for parameter-passing methods.

Course Outcome 4 (CO4):

1. Describe the role of a virtual method table in implementing dynamic method binding.
2. Identify one disadvantage of inheritance.

Course Outcome 5 (CO5): Evaluate the use of semaphores and monitors for providing competition synchronization and cooperation synchronization.

Model Question paper

QP CODE:

PAGES:3

Reg No: _____

Name : _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FOURTH SEMESTER B.TECH (MINOR) DEGREE EXAMINATION, MONTH &

YEAR Course Code: CST 282

Course Name: Programming Methodologies

Max.Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Differentiate between readability and writability.
2. Define binding and binding time
3. What are the advantages of user-defined enumeration types?
4. Define narrowing and widening conversions.
5. Why **for** statement in C language is more flexible than that of older languages?
6. What are the advantages and disadvantages of dynamic local variables in subprograms?
7. Explain the concept of dynamic method binding with an example.
8. Is it mandatory to use constructors in object oriented languages? Justify your answer.
9. What are the applications of logical programming languages?
10. Explain the working of *let* and *let-rec* constructs in Scheme.

Part B

Answer any one Question from each module. Each question carries 14 Marks

11.
 - (a) Explain different criteria used for evaluating languages. (7 marks)
 - (b) Explain the major methods of implementing programming languages. (7 marks)

OR

12.
 - (a) Explain the meanings, purposes, advantages and disadvantages of four categories of scalar variables according to their storage bindings. (7 marks)

- (b) What is referencing environment of a statement? Show the referencing environment at the indicated program points (1), (2), (3) & (4) for the following program segment. Assume that the programming language used is statically scoped.

program example;

var a, b : integer;

procedure sub1;

var x, y: integer;

begin { sub1 }

.....

end { sub1 }

(1)

procedure sub2;

var x : integer;

.....

procedure sub3;

var x: integer;

begin { sub3 }

.....

end { sub3 }

(2)

begin { sub2 }

.....

end { sub2 }

(3)

begin {example}

.....

end {example }

(4)

(7 Marks)

13.

- (a) Explain any two problems associated with the pointer data types and also indicate how dangling pointer problem can be solved.

(7 marks)

- (b) Describe the lazy and eager approaches for reclaiming garbage.

(7 marks)

OR

14.

- (a) What is meant by *side effect* and illustrate the advantages of referential transparency?

(8 marks)

- (b) Explain the terms: compound assignment operator, coercion and short circuit evaluation.

(6 marks)

15.

- (a) Explain different categories of iteration control statements.

(8 marks)

- (b) Explain techniques used for identifying correct referencing environment for a subprogram that was sent as a parameter.

(6 marks)

OR

16.

- (a) Describe the implementation models of Parameter passing.

(10 Marks)

- (b) Differentiate coroutines from conventional subprograms.

(4 marks)

17.

- (a) What is an exception handler? Explain how exceptions are handled in object oriented language?

(7 Marks)

- (b) What are the design issues in object oriented languages?

(7 Marks)

OR

18. Explain the following object oriented features:

- (i) Encapsulation
- (ii) Inheritance
- (iii) Constructors and Destructors
- (iv) Operator Overloading
- (v) Polymorphism

(14 Marks)

19.

- (a) Compare functional and imperative programming languages.

(7 Marks)

- (b) Explain the role of monitors in concurrency.

(7 Marks)

OR

20.

- (a) Explain the searching strategies used in Prolog. Why backward chaining is preferred over forward chaining in Prolog?

(10 Marks)

- (b) How does a binary semaphore differ from an ordinary semaphore?

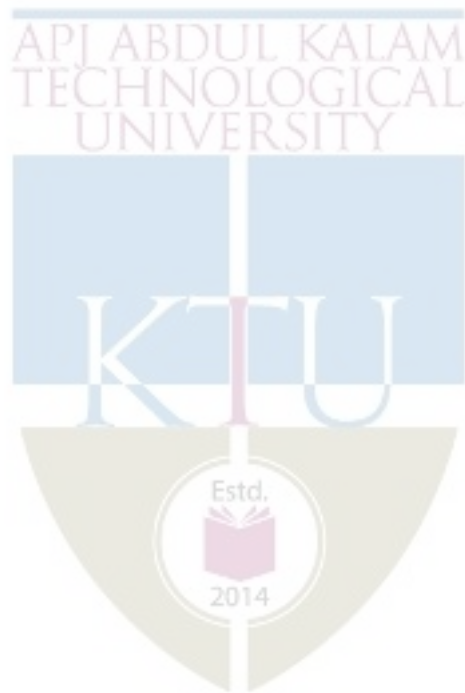
(4 Marks)

Teaching Plan

Module 1 (Introduction)		9 Hours
1.1	Introduction : Reasons for studying Concepts of programming languages, Programming Domains	1 Hour
1.2	Language Evaluation Criteria (Lecture 1)	1 Hour
1.3	Language Evaluation Criteria (Lecture 2)	1 Hour
1.4	Influence on Language Design, Language Design Trade-offs	1 Hour
1.5	Implementation Methods	1 Hour
1.6	Names, Variables	1 Hour
1.7	Concept of Binding	1 Hour
1.8	Scope and Lifetime	1 Hour
1.9	Referencing Environments	1 Hour
Module 2 (Data Types, Expressions and Assignment Statements)		8 Hours
2.1	Primitive Data Types, Character String Types	1 Hour
2.2	User-Defined Ordinal Types, Array Types	1 Hour
2.3	Record Types, List Types, Pointer and Reference Types	1 Hour
2.4	Implementation of pointer and reference types, Type Checking, Strong Typing, Type Equivalence	1 Hour
2.5	Expressions and Assignment Statements, Arithmetic Expressions	1 Hour
2.6	Overloaded Operators, Type Conversions	1 Hour
2.7	Relational and Boolean Expressions, Short-Circuit Evaluation	1 Hour
2.8	Assignment Statements, Mixed-mode Assignment	1 Hour
Module 3 (Statement Level Control Structures, Subprograms)		8 Hours
3.1	Selection Statements, Iterative Statements	1 Hour
3.2	Unconditional Branching	1 Hour

3.3	Guarded Commands	1 Hour
3.4	Subprograms: Design Issues of Subprograms	1 Hour
3.5	Local Referencing Environments	1 Hour
3.6	Parameter Passing Methods	1 Hour
3.7	Subprograms as Parameters, Overloaded Subprograms	1 Hour
3.8	Closures, Co-routines	1 Hour
Module 4 (Support for Object Oriented Programming, Exception Handling, Event handling)		10 Hours
4.1	Inheritance	1 Hour
4.2	Dynamic Binding	1 Hour
4.3	Design Issues for Object Oriented Languages	1 Hour
4.4	Support for Object Oriented Programming in C++	1 Hour
4.5	Implementation of Object-Oriented Constructs (Lecture 1)	1 Hour
4.6	Implementation of Object-Oriented Constructs (Lecture 2)	1 Hour
4.7	Implementation of Object-Oriented Constructs (Lecture 3)	1 Hour
4.8	Basic Concepts	1 Hour
4.9	Exception Handling - Design Issues	1 Hour
4.10	Introduction to Event Handling	1 Hour
Module 5 (Concurrency, Functional Programming Languages, Logic Programming languages)		10 Hours
5.1	Subprogram Level Concurrency	1 Hour
5.2	Semaphores	1 Hour
5.3	Monitors	1 Hour
5.4	Message Passing	1 Hour
5.5	Introduction to LISP and Scheme (Lecture 1)	1 Hour
5.6	Introduction to LISP and Scheme (Lecture 2)	1 Hour
5.7	Comparison of Functional and Imperative Languages	1 Hour
5.8	Basic Elements of Prolog (Lecture 1)	1 Hour

5.9	Basic Elements of Prolog (Lecture 2)	1 Hour
5.10	Applications of Logic Programming	1 Hour



CST 283	Python for Machine Learning	Category	L	T	P	Credit	Year of Introduction
		MINOR	3	1	0	4	2019

Preamble: This is a programming course for awarding B. Tech. Minor in Computer Science and Engineering with specialization in **Machine Learning**. The objective of the course is to provide learners an insight into Python programming, and develop programming skills to manage the development of software systems. It covers programming environment, important instructions, data representations, intermediate level features, Object Oriented Programming and file data processing of Python. This course lays the foundation to develop web applications, Machine Learning, and Artificial Intelligence-based applications and tools, Data Science and Data Visualization applications.

Prerequisite: Nil

Course Outcomes: After the completion of the course the student will be able to

CO1	Write, test and debug Python programs (Cognitive Knowledge level: Apply)
CO2	Illustrate uses of conditional (if, if-else, if-elif-else and switch-case) and iterative (while and for) statements in Python programs (Cognitive Knowledge level: Apply)
CO3	Develop programs by utilizing the modules Lists, Tuples, Sets and Dictionaries in Python (Cognitive Knowledge level: Apply)
CO4	Implement Object Oriented programs with exception handling (Cognitive Knowledge level: Apply)
CO5	Write programs in Python to process data stored in files by utilizing the modules Numpy, Matplotlib, and Pandas (Cognitive Knowledge level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓		✓						✓	✓
CO2	✓	✓	✓		✓					✓		✓
CO3	✓	✓	✓		✓	✓	✓					✓
CO4	✓	✓	✓		✓		✓					✓
CO5	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation

#PO	Broad PO	#PO	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (Marks in percentage)	Test 2 (Marks in percentage)	End Semester Examination (Marks in percentage)
Remember	20	20	20
Understand	35	35	35
Apply	45	45	45
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : 10 marks
 Continuous Assessment Test : 25 marks
 Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. The first series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 sub-divisions and carries 14 marks.

SYLLABUS

Module I

Programming Environment and Python Basics:

Getting Started with Python Programming - Running code in the interactive shell, Editing, Saving, and Running a script. Using editors - IDLE, Jupyter. The software development process - Case Study.

Basic coding skills - Working with data types, Numeric data types and Character sets, Keywords, Variables and Assignment statement, Operators, Expressions, Working with numeric data, Type conversions, Comments in the program. Input, Processing, and Output. Formatting output. How Python works. Detecting and correcting syntax errors. Using built in functions and modules in math module.

Module II

Building Python Programs:

Control statements - Selection structure (if-else, switch-case), Iteration structure (for, while), Testing the control statements, Lazy evaluation. Functions - Hiding redundancy and complexity, Arguments and return values, Variable scopes and parameter passing, Named arguments, Main function, Working with recursion, Lambda functions. Strings and number systems - String function, Handling numbers in various formats.

Module III

Data Representation:

Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension. Work with tuples. Sets. Work with dates and times. Dictionaries - Dictionary

functions, dictionary literals, adding and removing keys, accessing and replacing values, traversing dictionaries, reverse lookup. Case Study - Data Structure Selection.

Module IV

Object Oriented Programming:

Design with classes - Objects and Classes, Methods, Instance Variables, Constructor, Accessors and Mutators. Structuring classes with Inheritance and Polymorphism. Abstract Classes. Exceptions - Handle a single exception, handle multiple exceptions.

Module V

Data Processing:

The *os* and *sys* modules. Introduction to file I/O - Reading and writing text files, Manipulating binary files. NumPy - Basics, Creating arrays, Arithmetic, Slicing, Matrix Operations, Random numbers. Plotting and visualization. Matplotlib - Basic plot, Ticks, Labels, and Legends. Working with CSV files. – Pandas - Reading, Manipulating, and Processing Data.

Text Books:

1. Kenneth A Lambert., Fundamentals of Python : First Programs, 2/e, Cengage Publishing, 2016
2. Wes McKinney, Python for Data Analysis, 2/e, Shroff / O'Reilly Publishers, 2017

Reference Books:

1. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2/e, Schroff, 2016
2. Michael Urban and Joel Murach, Python Programming, Shroff/Murach, 2016
3. David M.Baezly, Python Essential Reference. Addison-Wesley Professional; 4/e, 2009.
4. Charles Severance. Python for Informatics: Exploring Information,
5. <http://swcarpentry.github.io/python-novice-gapminder/>

Sample Course Level Assessment Questions

Course Outcome1(CO1): What is type conversion? How is it done in Python?

Course Outcome 2(CO2): Write a Python program which takes a positive integer **n** as input and finds the sum of cubes all positive even numbers less than or equal to the number.

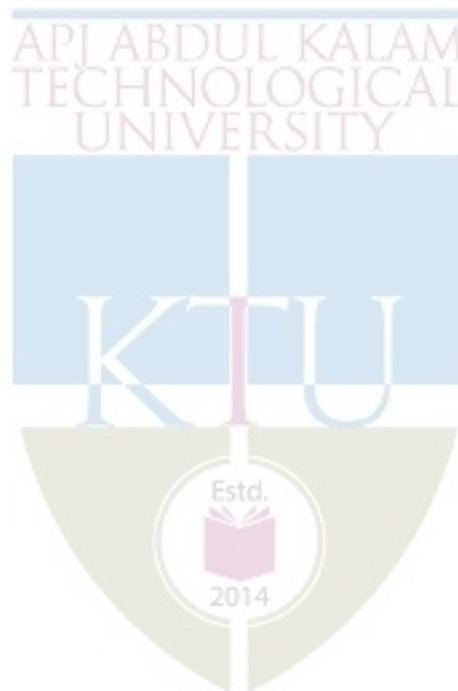
Course Outcome 3(CO3): Given is a list of words, *wordlist*, and a string, *name*. Write a Python function which takes *wordlist* and *name* as input and returns a tuple. The first element of

the output tuple is the number of words in the *wordlist* which have *name* as a substring in it. The second element of the tuple is a list showing the index at which the *name* occurs in each of the words of the *wordlist* and a 0 if it doesn't occur.

Course Outcome 4(CO4): Write a Python program to implement the addition, subtraction, and multiplication of complex numbers using classes. Use constructors to create objects. The input to the program consist of real and imaginary parts of the complex numbers.

Course Outcome 5(CO5): Given a file “auto.csv” of automobile data with the fields *index*, *company*, *body-style*, *wheel-base*, *length*, *engine-type*, *num-of-cylinders*, *horsepower*, *average-mileage*, and *price*, write python code to

- 1) Clean and Update the CSV file
- 2) Print total cars of all companies
- 3) Find the average mileage of all companies
- 4) Find the highest priced car of all companies.



Model Question Paper

QP CODE:

PAGES:

Reg No: _____

Name: _____

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
THIRD SEMESTER B.TECH (MINOR) DEGREE EXAMINATION, MONTH & YEAR**

Course Code: CST 283

Course name : PYTHON FOR MACHINE LEARNING

Max Marks: 100

Duration: 3 Hours

PART-A

(Answer All Questions. Each question carries 3 marks)

1. Explain the basic data types available in Python, with examples.
2. Write a Python program to reverse a number and also find the sum of digits of the number. Prompt the user for input.
3. Explain the concept of scope and lifetime of variables in Python programming language, with a suitable example.
4. Discuss format specifiers and escape sequences with examples.
5. Discuss the relation between tuples, lists, and dictionaries in detail.
6. Discuss the following dictionary methods with an example.
i. *get()* ii. *Keys()* iii. *pop()* iv. *update()* v. *values()* vi. *items()*
7. What is polymorphism? Give an example in the context of OOP in Python.
8. How is exception handling accomplished in Python programs?
9. Write a note on the **os** and **os.path** modules in Python. Also, discuss the *walk()* and *getcwd()* methods of the **os** module.
10. Describe the characteristics of the CSV format.

PART-B

(Answer any one full question from each module)

11. (a) Compare and contrast interpreted languages and compiled languages. How does it affect the quality of program development and execution of the program? (6)
- (b) What are the possible errors in a Python program. Write a Python program to print the value of $2^{2n}+n+5$ for n provided by the user. (8)

OR

12. (a) Describe Arithmetic operators, Assignment operators, Comparison operators, Logical operators, and Bitwise operators in detail with examples. (6)
- (b) Explain the software development process in detail. (8)
13. (a) Write a Python code to check whether a given year is a leap year or not [An year is a leap year if it's divisible by 4 but not divisible by 100 except for those divisible by 400]. (5)
- (b) Input 4 integers (+ve and -ve). Write a Python code to find the sum of negative numbers, positive numbers, and print them. Also, find the averages of these two groups of numbers and print. (9)

OR

14. (a) Write a Python program to find the value for $\sin(x)$ up to n terms using the series (8)

$$\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad \text{where } x \text{ is in degrees}$$

- (b) Write a Python code to determine whether the given string is a Palindrome or not using slicing. Do not use any string function. (6)
15. (a) Write a Python code to create a function called *list_of_frequency* that takes a string and prints the letters in non-increasing order of the frequency of their occurrences. Use dictionaries. (5)
- (b) Write a Python program to read a list of numbers and sort the list in a non-decreasing order without using any built in functions. Separate function should be written to sort the list wherein the name of the list is passed as the parameter. (9)

OR

16. (a) Illustrate the following Set methods with an example. (6)
i. *intersection()* ii. *Union()* iii. *Issubset()* iv. *Difference()* v. *update()* vi. *discard()*

- (b) Write a Python program to check the validity of a password given by the user. (8)

The Password should satisfy the following criteria:

1. Contains at least one letter between **a** and **z**
2. Contains at least one number between **0** and **9**
3. Contains at least one letter between **A** and **Z**
4. Contains at least one special character from **!, @, #, \$, %, ^, &, ***
5. Minimum length of password: **6**

17. (a) How can a class be instantiated in Python? Write a Python program to express the instances as return values to define a class RECTANGLE with parameters *height*, *width*, *corner_x*, and *corner_y* and member functions to find center, area, and perimeter of an instance. (10)

- (b) Explain inheritance in Python. Give examples for each type of inheritance. (4)

OR

18. (a) Write a Python class named **Circle** constructed by a radius and two methods which will compute the area and the perimeter of a given circle (6)

- (b) Write Python program to create a class called as **Complex** and implement *__add__()* method to add two complex numbers. Display the result by overloading the + Operator. (8)

19. (a) Write a Python program to add two matrices and also find the transpose of the resultant matrix. (8)

- (b) Given a file "auto.csv" of automobile data with the fields *index*, *company*, *body-style*, *wheel-base*, *length*, *engine-type*, *num-of-cylinders*, *horsepower*, *average-mileage*, and *price*, write Python codes using Pandas to (6)
- 1) Clean and Update the CSV file
 - 2) Print total cars of all companies
 - 3) Find the average mileage of all companies
 - 4) Find the highest priced car of all companies.

OR

20. (a) Write Python program to write the data given below to a CSV file. (5)

SN	Name	Country	Contribution	Year	
1	Linus Torvalds	Finland	Linux Kernel	1991	
2	Tim Berners-Lee	England	World Wide Web	1990	
3	Guido van Rossum	Netherlands	Python	1991	

- (b) Given the sales information of a company as CSV file with the following fields *month_number*, *facecream*, *facewash*, *toothpaste*, *bathingssoap*, *shampoo*, *moisturizer*, *total_units*, *total_profit*. Write Python codes to visualize the data as follows (9)
- 1) Toothpaste sales data of each month and show it using a scatter plot
 - 2) Face cream and face wash product sales data and show it using the bar chart
 - 3) Calculate total sale data for last year for each product and show it using a Pie chart.

(14X5=70)

Teaching Plan

Module 1: Programming Environment and Python Basics		(10 hours)
1.1	Getting Started with Python Programming: Running code in the interactive shell Editing, Saving, and Running a script	1 hour
1.2	Using editors: IDLE	1 hour
1.3	Jupyter	1 hour
1.4	The software development process: Case Study.	1 hour
1.5	Basic coding skills: Working with data types, Numeric data types and Character sets, Keywords, Variables and Assignment statement, Operators, Expressions,	1 hour
1.6	Working with numeric data, Type conversions, Comments in the program	1 hour
1.7	Input, Processing, and Output, Formatting output – How Python works	1 hour
1.8	How Python works – Detecting and correcting syntax errors	1 hour
1.9	Using built in functions and modules: Case – Using math module	1 hour
1.10	Using built in functions and modules: Case – Using math module (Examples)	1 hour

Module 2: Building Python Programs		(8 hours)
2.1	Control statements: Selection structure (if-else, switch-case),	1 hour
2.2	Iteration structure(for, while), Testing the control statements, Lazy evaluation	1 hour
2.3	Functions: Hiding redundancy and complexity, Arguments and return values,	1 hour
2.4	Variable scopes and parameter passing	1 hour
2.5	Named arguments, Main function,	1 hour
2.6	Working with recursion, Lambda functions	1 hour
2.7	Strings and number systems: String function	1 hour
2.8	Handling numbers in various format	1 hour
Module 3: Data Representation		(9 hours)
3.1	Lists: Basic list Operations and functions, List of lists	1 hour
3.2	Slicing, Searching and sorting list	1 hour
3.3	List comprehension	1 hour
3.4	Work with tuples, Sets	1 hour
3.5	Work with dates and times	1 hour
3.6	Dictionaries: Dictionary functions,	1 hour
3.7	Dictionary literals, adding and removing keys, accessing & replacing values	1 hour
3.8	Traversing dictionaries, reverse lookup	1 hour
3.9	Case Study: Data Structure Selection	1 hour
Module 4: Object Oriented Programming		(8 hours)
4.1	Design with classes : Objects and Classes, Methods, Instance Variables	1 hour
4.2	Constructor, Accessors and Mutators	1 hour
4.3	Structuring classes with Inheritance	1 hour
4.4	Polymorphism	1 hour
4.5	Abstract Classes	1 hour
4.6	Abstract Classes	1 hour
4.7	Exceptions : Handle a single exception	1 hour

4.8	handle multiple exceptions	1 hour
Module 5: Data Processing		(10 hours)
5.1	The <i>os</i> and <i>sys</i> modules	1 hour
5.2	Introduction to file I/O: Reading and writing text files	1 hour
5.3	Manipulating binary files	1 hour
5.4	NumPy : Basics, Creating arrays, Arithmetic, Slicing	1 hour
5.5	Matrix Operations, Random numbers.	1 hour
5.6	Matplotlib : Basic plot	1 hour
5.7	Matplotlib - Ticks, Labels, and Legends	1 hour
5.8	Working with CSV files	1 hour
5.9	Pandas : Reading, Manipulating	1 hour
5.10	Pandas : Processing Data and Visualize.	1 hour



CODE CST 284	Mathematics for Machine Learning	CATEGORY	L	T	P	CREDIT
		MINOR	3	1	0	4

Preamble: This is a foundational course for awarding B. Tech. Minor in Computer Science and Engineering with specialization in **Machine Learning**. The purpose of this course is to introduce mathematical foundations of basic Machine Learning concepts among learners, on which Machine Learning systems are built. This course covers Linear Algebra, Vector Calculus, Probability and Distributions, Optimization and Machine Learning problems. Concepts in this course help the learners to understand the mathematical principles in Machine Learning and aid in the creation of new Machine Learning solutions, understand & debug existing ones, and learn about the inherent assumptions & limitations of the current methodologies.

Prerequisite:

1. A sound background in higher secondary school Mathematics.
2. Python for Machine Learning (CST 253)

Course Outcomes: After the completion of the course the student will be able to

CO 1	Make use of the concepts, rules and results about linear equations, matrix algebra, vector spaces, eigenvalues & eigenvectors and orthogonality & diagonalization to solve computational problems (Cognitive Knowledge Level: Apply)
CO 2	Perform calculus operations on functions of several variables and matrices, including partial derivatives and gradients (Cognitive Knowledge Level: Apply)
CO 3	Utilize the concepts, rules and results about probability, random variables, additive & multiplicative rules, conditional probability, probability distributions and Bayes' theorem to find solutions of computational problems (Cognitive Knowledge Level: Apply)
CO 4	Train Machine Learning Models using unconstrained and constrained optimization methods (Cognitive Knowledge Level: Apply)
CO 5	Illustrate how the mathematical objects - linear algebra, probability, and calculus can be used to design machine learning algorithms (Cognitive Knowledge Level: Understand)

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1	√	√	√	√								√
CO 2	√	√	√									√
CO 3	√	√	√	√								√
CO 4	√	√	√	√		√						√
CO 5	√	√	√	√	√	√				√		√

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember	20%	20%	20%
Understand	40%	40%	40%
Apply	40%	40%	40%
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

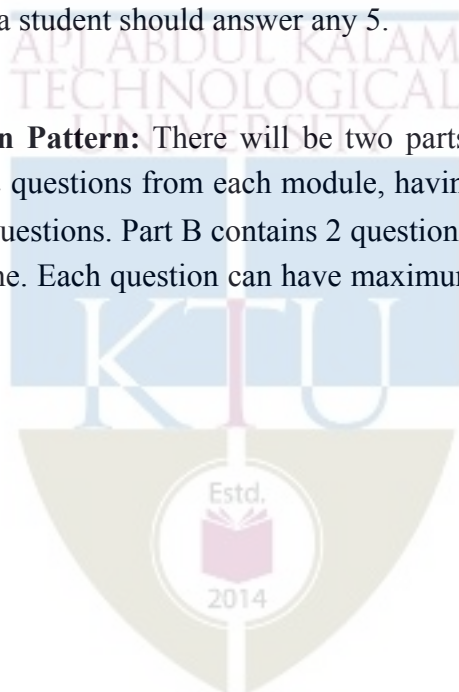
Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which student should answer anyone. Each question can have maximum 2 sub-divisions and carries 14 marks.



Syllabus

Module 1

LINEAR ALGEBRA : Systems of Linear Equations – Matrices, Solving Systems of Linear Equations. Vector Spaces - Linear Independence, Basis and Rank, Linear Mappings, Norms, - Inner Products - Lengths and Distances - Angles and Orthogonality - Orthonormal Basis - Orthogonal Complement - Orthogonal Projections. Matrix Decompositions - Determinant and Trace, Eigenvalues and Eigenvectors, Cholesky Decomposition, Eigen decomposition and Diagonalization, Singular Value Decomposition, Matrix Approximation.

Module 2

VECTOR CALCULUS : Differentiation of Univariate Functions - Partial Differentiation and Gradients, Gradients of Vector Valued Functions, Gradients of Matrices, Useful Identities for Computing Gradients. Back propagation and Automatic Differentiation - Higher Order Derivatives- Linearization and Multivariate Taylor Series.

Module 3

Probability and Distributions : Construction of a Probability Space - Discrete and Continuous Probabilities, Sum Rule, Product Rule, and Bayes' Theorem. Summary Statistics and Independence – Important Probability distributions - Conjugacy and the Exponential Family - Change of Variables/Inverse Transform.

Module 4

Optimization : Optimization Using Gradient Descent - Gradient Descent With Momentum, Stochastic Gradient Descent. Constrained Optimization and Lagrange Multipliers - Convex Optimization - Linear Programming - Quadratic Programming.

Module 5

CENTRAL MACHINE LEARNING PROBLEMS : Data and Learning Model- Empirical Risk Minimization - Parameter Estimation - Directed Graphical Models.

Linear Regression - Bayesian Linear Regression - Maximum Likelihood as Orthogonal Projection.

Dimensionality Reduction with Principal Component Analysis - Maximum Variance Perspective, Projection Perspective. Eigenvector Computation and Low Rank Approximations.

Density Estimation with Gaussian Mixture Models - Gaussian Mixture Model, Parameter Learning via Maximum Likelihood, EM Algorithm.

Classification with Support Vector Machines - Separating Hyperplanes, Primal Support Vector Machine, Dual Support Vector Machine, Kernels.

Text book:

1. Mathematics for Machine Learning by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong published by Cambridge University Press (freely available at <https://mml-book.github.io>)

Reference books:

1. Linear Algebra and Its Applications, 4th Edition by Gilbert Strang
2. Linear Algebra Done Right by Axler, Sheldon, 2015 published by Springer
3. Introduction to Applied Linear Algebra by Stephen Boyd and Lieven Vandenberghe, 2018 published by Cambridge University Press
4. Convex Optimization by Stephen Boyd and Lieven Vandenberghe, 2004 published by Cambridge University Press
5. Pattern Recognition and Machine Learning by Christopher M Bishop, 2006, published by Springer
6. Learning with Kernels – Support Vector Machines, Regularization, Optimization, and Beyond by Bernhard Scholkopf and Smola, Alexander J Smola, 2002, published by MIT Press
7. Information Theory, Inference, and Learning Algorithms by David J. C MacKay, 2003 published by Cambridge University Press
8. Machine Learning: A Probabilistic Perspective by Kevin P Murphy, 2012 published by MIT Press.
9. The Nature of Statistical Learning Theory by Vladimir N Vapnik, 2000, published by Springer

Sample Course Level Assessment Questions.

Course Outcome 1 (CO1):

1. Find the set \mathcal{S} of all solutions in \mathbf{x} of the following inhomogeneous linear systems $\mathbf{Ax} = \mathbf{b}$, where \mathbf{A} and \mathbf{b} are defined as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 3 \\ 6 \\ 5 \\ -1 \end{bmatrix}$$

2. Determine the inverses of the following matrix if possible

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

3. Are the following sets of vectors linearly independent?

$$\mathbf{x}_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

4. A set of n linearly independent vectors in \mathbf{R}^n forms a basis. Does the set of vectors $(2, 4, -3)$, $(0, 1, 1)$, $(0, 1, -1)$ form a basis for \mathbf{R}^3 ? Explain your reasons.
5. Consider the transformation $T(\mathbf{x}, \mathbf{y}) = (\mathbf{x} + \mathbf{y}, \mathbf{x} + 2\mathbf{y}, 2\mathbf{x} + 3\mathbf{y})$. Obtain $\ker T$ and use this to calculate the nullity. Also find the transformation matrix for T .
6. Find the characteristic equation, eigenvalues, and eigenspaces corresponding to each eigenvalue of the following matrix

$$\begin{bmatrix} 2 & 0 & 4 \\ 0 & 3 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

7. Diagonalize the following matrix, if possible

$$\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 3 \end{bmatrix}$$

8. Find the singular value decomposition (SVD) of the following matrix

$$\begin{bmatrix} 0 & 1 & 1 \\ \sqrt{2} & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Course Outcome 2 (CO2):

1. For a scalar function $f(x, y, z) = x^2 + 3y^2 + 2z^2$, find the gradient and its magnitude at the point $(1, 2, -1)$.
2. Find the maximum and minimum values of the function $f(x, y) = 4x + 4y - x^2 - y^2$ subject to the condition $x^2 + y^2 \leq 2$.
3. Suppose you were trying to minimize $f(x, y) = x^2 + 2y + 2y^2$. Along what vector should you travel from $(5, 12)$?
4. Find the second order Taylor series expansion for $f(x, y) = (x + y)^2$ about $(0, 0)$.
5. Find the critical points of $f(x, y) = x^2 - 3xy + 5x - 2y + 6y^2 + 8$.
6. Compute the gradient of the Rectified Linear Unit (ReLU) function $\text{ReLU}(z) = \max(0, z)$.
7. Let $L = \|Ax - b\|_2^2$, where A is a matrix and x and b are vectors. Derive dL in terms of dx .

Course Outcome 3 (CO3):

1. Let J and T be independent events, where $P(J)=0.4$ and $P(T)=0.7$.
 - i. Find $P(J \cap T)$
 - ii. Find $P(J \cup T)$
 - iii. Find $P(J \cap T')$
2. Let A and B be events such that $P(A)=0.45$, $P(B)=0.35$ and $P(A \cup B)=0.5$. Find $P(A|B)$.
3. A random variable R has the probability distribution as shown in the following table:

r	1	2	3	4	5
$P(R=r)$	0.2	a	b	0.25	0.15

- i. Given that $E(R)=2.85$, find a and b .
 - ii. Find $P(R>2)$.
4. A biased coin (with probability of obtaining a head equal to $p > 0$) is tossed repeatedly and independently until the first head is observed. Compute the probability that the first head appears at an even numbered toss.
5. Two players A and B are competing at a trivia quiz game involving a series of questions. On any individual question, the probabilities that A and B give the correct answer are p and q respectively, for all questions, with outcomes for different questions being independent. The game finishes when a player wins by answering a question correctly. Compute the probability that A wins if
 - i. A answers the first question,
 - ii. B answers the first question.
6. A coin for which $P(\text{heads}) = p$ is tossed until two successive tails are obtained. Find the probability that the experiment is completed on the n^{th} toss.
7. You roll a fair dice twice. Let the random variable X be the product of the outcomes of the two rolls. What is the probability mass function of X ? What are the expected value and the standard deviation of X ?

8. While watching a game of Cricket, you observe someone who is clearly supporting Mumbai Indians. What is the probability that they were actually born within 25KM of Mumbai? Assume that:
- the probability that a randomly selected person is born within 25KM of Mumbai is $1/20$;
 - the chance that a person born within 25KMs of Mumbai actually supports MI is $7/10$;
 - the probability that a person not born within 25KM of Mumbai supports MI with probability $1/10$.
9. What is an exponential family? Why are exponential families useful?
10. Let Z_1 and Z_2 be independent random variables each having the standard normal distribution. Define the random variables X and Y by $X = Z_1 + 3Z_2$ and $Y = Z_1 + Z_2$. Argue that the joint distribution of (X, Y) is a bivariate normal distribution. What are the parameters of this distribution?
11. Given a continuous random variable x , with cumulative distribution function $F_x(x)$, show that the random variable $y = F_x(x)$ is uniformly distributed.
12. Explain Normal distribution, Binomial distribution and Poisson distribution in the exponential family form.

Course Outcome 4(CO4):

1. Find the extrema of $f(x, y) = x$ subject to $g(x, y) = x^2 + 2y^2 = 3$.
2. Maximize the function $f(x, y, z) = xy + yz + xz$ on the unit sphere $g(x, y, z) = x^2 + y^2 + z^2 = 1$.
3. Provide necessary and sufficient conditions under which a quadratic optimization problem be written as a linear least squares problem.
4. Consider the univariate function $f(x) = x^3 + 6x^2 - 3x - 5$. Find its stationary points and indicate whether they are maximum, minimum, or saddle points.
5. Consider the update equation for stochastic gradient descent. Write down the update when we use a mini-batch size of one.

6. Consider the function

$$f(x) = (x_1 - x_2)^2 + \frac{1}{1 + x_1^2 + x_2^2}.$$

- i. Is $f(x)$ a convex function? Justify your answer.
 - ii. Is $(1, -1)$ a local/global minimum? Justify your answer.
7. Is the function $f(x, y) = 2x^2 + y^2 + 6xy - x + 3y - 7$ convex, concave, or neither? Justify your answer.
8. Consider the following convex optimization problem

$$\text{minimize } \frac{x^2}{2} + x + 4y^2 - 2y$$

Subject to the constraint $x + y \geq 4, x, y \geq 1$.

Derive an explicit form of the Lagrangian dual problem.

9. Solve the following LP problem with the simplex method.

$$\text{max } 5x_1 + 6x_2 + 9x_3 + 8x_4$$

subject to the constraints

$$\begin{aligned} x_1 + 2x_2 + 3x_3 + x_4 &\leq 5 \\ x_1 + x_2 + 2x_3 + 3x_4 &\leq 3 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

Course Outcome 5 (CO5):

1. What is a loss function? Give examples.
2. What are training/validation/test sets? What is cross-validation? Name one or two examples of cross-validation methods.
3. Explain generalization, overfitting, model selection, kernel trick, Bayesian learning

4. Distinguish between Maximum Likelihood Estimation (MLE) and Maximum A Posteriori Estimation (MAP)?
5. What is the link between structural risk minimization and regularization?
6. What is a kernel? What is a dot product? Give examples of kernels that are valid dot products.
7. What is ridge regression? How can one train a ridge regression linear model?
8. What is Principal Component Analysis (PCA)? Which eigen value indicates the direction of largest variance? In what sense is the representation obtained from a projection onto the eigen directions corresponding to the largest eigen values optimal for data reconstruction?
9. Suppose that you have a linear support vector machine (SVM) binary classifier. Consider a point that is currently classified correctly, and is far away from the decision boundary. If you remove the point from the training set, and re-train the classifier, will the decision boundary change or stay the same? Explain your answer in one sentence.
10. Suppose you have n independent and identically distributed (i.i.d) sample data points $\mathbf{x}_1, \dots, \mathbf{x}_n$. These data points come from a distribution where the probability of a given datapoint \mathbf{x} is

$$P(\mathbf{x}) = \frac{1}{\theta} e^{-\frac{1}{\theta} \mathbf{x}}.$$

Prove that the MLE estimate of parameter θ is the sample mean.

11. Suppose the data set y_1, \dots, y_n is drawn from a random sample consisting of i.i.d. discrete uniform distributions with range 1 to N . Find the maximum likelihood estimate of N .
12. Ram has two coins: one fair coin and one biased coin which lands heads with probability $3/4$. He picks one coin at random (50-50) and flips it repeatedly until he gets a tails. Given that he observes 3 heads before the first tails, find the posterior probability that he picked each coin.
 - i. What are the prior and posterior odds for the fair coin?
 - ii. What are the prior and posterior predictive probabilities of heads on the next flip? Here prior predictive means prior to considering the data of the first four flips.

Model Question paper

QP Code :

Total Pages: 4

Reg No.: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

IV SEMESTER B.TECH (MINOR) DEGREE EXAMINATION, MONTH and YEAR

Course Code: CST 284

Course Name: MATHEMATICS FOR MACHINE LEARNING

Max. Marks: 100

Duration: 3 Hours

PART A

Answer all questions, each carries 3 marks.

Marks

- 1 Show that with the usual operation of scalar multiplication but with addition on reals given by $x \# y = 2(x + y)$ is not a vector space.
- 2 Find the eigenvalues of the following matrix in terms of k . Can you find an eigenvector corresponding to each of the eigenvalues?
$$\begin{bmatrix} 1 & k \\ 2 & 1 \end{bmatrix}$$
- 3 Let $f(x, y, z) = xye^r$, where $r = x^2 + z^2 - 5$. Calculate the gradient of f at the point $(1, 3, -2)$.
- 4 Compute the Taylor polynomials T_n , $n = 0, \dots, 5$ of $f(x) = \sin(x) + \cos(x)$ at $x_0 = 0$.
- 5 Let X be a continuous random variable with probability density function on $0 \leq x \leq 1$ defined by $f(x) = 3x^2$. Find the pdf of $Y = X^2$.
- 6 Show that if two events A and B are independent, then A and B' are independent.
- 7 Explain the principle of the gradient descent algorithm.

- 8 Briefly explain the difference between (batch) gradient descent and stochastic gradient descent. Give an example of when you might prefer one over the other.
- 9 What is the empirical risk? What is “empirical risk minimization”?
- 10 Explain the concept of a Kernel function in Support Vector Machines. Why are kernels so useful? What properties a kernel should possess to be used in an SVM?

PART B

Answer any one Question from each module. Each question carries 14 Marks

- 11 a) i. Find all solutions to the system of linear equations (6)
- $$\begin{aligned} -4x + 5z &= -2 \\ -3x - 3y + 5z &= 3 \\ -x + 2y + 2z &= -1 \end{aligned}$$
- ii. Prove that all vectors orthogonal to $[2, -3, 1]^T$ form a subspace W of \mathbb{R}^3 . What is $\dim(W)$ and why?
- b) Use the Gram-Schmidt process to find an orthogonal basis for the column space of the following matrix (8)

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

OR

- 12 a) i. Let L be the line through the origin in \mathbf{R}^2 that is parallel to the vector

$[3, 4]^T$. Find the standard matrix of the orthogonal projection onto L . Also find the point on L which is closest to the point $(7, 1)$ and find the point on L which is closest to the point $(-3, 5)$.

- ii. Find the rank-1 approximation of

$$\begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$$

- b) i. Find an orthonormal basis of \mathbf{R}^3 consisting of eigenvectors for the following matrix

$$\begin{bmatrix} 1 & 0 & -2 \\ 0 & 5 & 0 \\ -2 & 0 & 4 \end{bmatrix}$$

- ii. Find a 3×3 orthogonal matrix S and a 3×3 diagonal matrix D such that $A = SDS^T$.

- 13 a) A skier is on a mountain with equation $z = 100 - 0.4x^2 - 0.3y^2$, where z denotes height. (8)

- i. The skier is located at the point with xy -coordinates $(1, 1)$, and wants to ski downhill along the steepest possible path. In which direction (indicated by a vector (a, b) in the xy -plane) should the skier begin skiing.

- ii. The skier begins skiing in the direction given by the xy -vector (a, b) you found in part (i), so the skier heads in a direction in space given by the vector (a, b, c) . Find the value of c .

- b) Find the linear approximation to the function $f(x, y) = 2 - \sin(-x - 3y)$ at the point $(0, \pi)$, and then use your answer to estimate $f(0.001, \pi)$. (6)

OR

- 14 a) Let g be the function given by (8)

$$g(x, y) = \begin{cases} \frac{x^2 y}{x^2 + y^2} & \text{if } (x, y) \neq (0, 0); \\ 0 & \text{if } (x, y) = (0, 0). \end{cases}$$

- i. Calculate the partial derivatives of g at $(0, 0)$.
 - ii. Show that g is not differentiable at $(0, 0)$.
- b) Find the second order Taylor series expansion for $f(x, y) = e^{-(x^2+y^2)} \cos(xy)$ (6)
about $(0, 0)$.
- 15 a) There are two bags. The first bag contains four mangos and two apples; (6)
the second bag contains four mangos and four apples. We also have a
biased coin, which shows “heads” with probability 0.6 and “tails” with
probability 0.4. If the coin shows “heads”. we pick a fruit at
random from bag 1; otherwise we pick a fruit at random from bag 2. Your
friend flips the coin (you cannot see the result), picks a fruit at random
from the corresponding bag, and presents you a mango.
What is the probability that the mango was picked from bag 2?
- b) Suppose that one has written a computer program that sometimes (8)
compiles and sometimes not (code does not change). You decide to model
the apparent stochasticity (success vs. no success) x of the compiler using
a Bernoulli distribution with parameter μ :

$$p(x | \mu) = \mu^x (1 - \mu)^{1-x}, \quad x \in \{0, 1\}$$

Choose a conjugate prior for the Bernoulli likelihood and compute the
posterior distribution $p(\mu | x_1, \dots, x_N)$.

OR

- 16 a) Consider a mixture of two Gaussian distributions (8)

$$0.4\mathcal{N}\left(\begin{bmatrix} 10 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) + 0.6\mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 8.4 & 2.0 \\ 2.0 & 1.7 \end{bmatrix}\right)$$

- i. Compute the marginal distributions for each dimension.
 - ii. Compute the mean, mode and median for each marginal distribution.
 - iii. Compute the mean and mode for the two-dimensional distribution.
- b) Express the Binomial distribution as an exponential family distribution. (6)
Also express the Beta distribution as an exponential family distribution.
Show that the product of the Beta and the Binomial distribution is also a member of the exponential family.
- 17 a) Find the extrema of $f(x,y,z) = x - y + z$ subject to $g(x,y,z) = x^2 + y^2 + z^2 = 2$. (8)
- b) Let

$$P = \begin{bmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{bmatrix}, q = \begin{bmatrix} -22.0 \\ -14.5 \\ 13.0 \end{bmatrix}, \text{ and } r = 1.$$

Show that $x^* = (1, 1/2, -1)$ is optimal for the optimization problem

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Px + q^T x + r \\ \text{s.t.} \quad & -1 \leq x_i \leq 1, i = 1, 2, 3. \end{aligned} \quad (6)$$

OR

- 18 a) Derive the gradient descent training rule assuming that the target function (8)
is represented as $o_d = w_0 + w_1x_1 + \dots + w_nx_n$. Define explicitly the cost/error function E , assuming that a set of training examples D is provided, where each training example $d \in D$ is associated with the target output t_d .

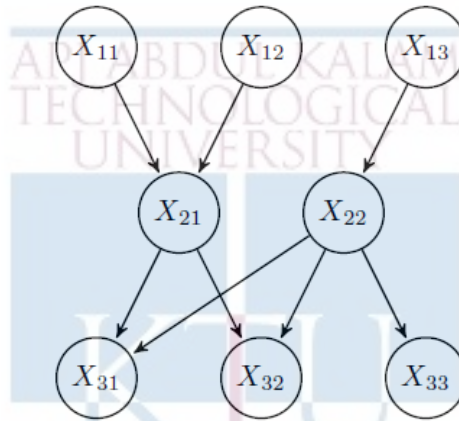
- b) Find the maximum value of $f(x,y,z) = xyz$ given that $g(x,y,z) = x + y + z = 3$ and $x,y,z \geq 0$. (6)

- 19 a) Consider the following probability distribution (7)

$$P_{\theta}(x) = 2\theta x e^{-\theta x^2}$$

where θ is a parameter and x is a positive real number. Suppose you get m i.i.d. samples x_i drawn from this distribution. Compute the maximum likelihood estimator for θ based on these samples.

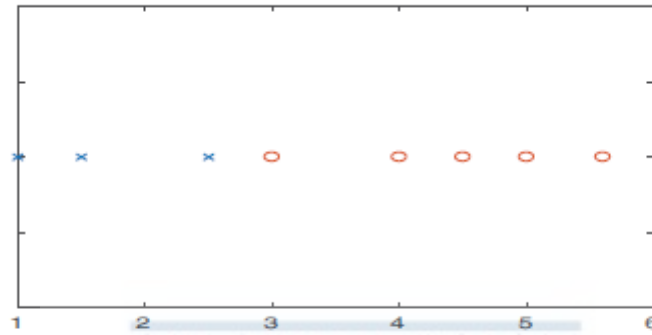
- b) Consider the following Bayesian network with boolean variables. (7)



- i. List variable(s) conditionally independent of X_{33} given X_{11} and X_{12}
- ii. List variable(s) conditionally independent of X_{33} and X_{22}
- iii. Write the joint probability $P(X_{11}, X_{12}, X_{13}, X_{21}, X_{22}, X_{31}, X_{32}, X_{33})$ factored according to the Bayes net. How many parameters are necessary to define the conditional probability distributions for this Bayesian network?
- iv. Write an expression for $P(X_{13} = 0, X_{22} = 1, X_{33} = 0)$ in terms of the conditional probability distributions given in your answer to part (iii). Justify your answer.

OR

- 20 a) Consider the following one dimensional training data set, 'x' denotes negative examples and 'o' positive examples. The exact data points and their labels are given in the table below. Suppose a SVM is used to classify this data. (6)



x	1	1.5	2.5	3	4	4.5	5	5.6
y	-1	-1	-1	1	1	1	1	1

- Indicate which are the support vectors and mark the decision boundary.
- Give the value of the cost function and the model parameter after training.



b) Suppose that we are fitting a Gaussian mixture model for data (8)

items consisting of a single real value, x , using $K = 2$ components.

We have $N = 5$ training cases, in which the values of x are as **5, 15, 25, 30, 40**. Using the EM algorithm to find the maximum likelihood estimates for the model parameters, what are the mixing proportions for the two components, π_1 and π_2 , and the means for the two components, μ_1 and μ_2 . The standard deviations for the two components are fixed at 10.

Suppose that at some point in the EM algorithm, the **E** step found that the responsibilities of the two components for the five data items were as follows:

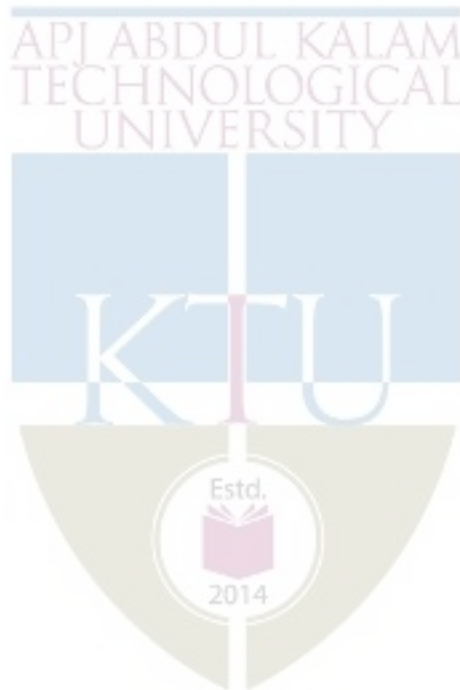
	r_{i1}	r_{i2}
	0.2	0.8
	0.2	0.8
	0.8	0.2
	0.9	0.1
	0.9	0.1

What values for the parameters π_1, π_2, μ_1 , and μ_2 will be found in the next **M** step of the algorithm?

Teaching Plan		
No	Topic	No. of Lectures (45)
1	Module-I (LINEAR ALGEBRA)	8
1.	Systems of Linear Equations – Matrices, Solving Systems of Linear Equations. Vector Spaces - Linear Independence.	1
2.	Vector Spaces - Basis and Rank	1
3.	Linear Mappings	1
4.	Norms, Inner Products, Lengths and Distances, Angles and Orthogonality, Orthonormal Basis, Orthogonal Complement	1
5.	Orthogonal Projections, Matrix Decompositions, Determinant and Trace.	1
6.	Eigenvalues and Eigenvectors	1
7.	Cholesky Decomposition, Eigen decomposition and Diagonalization	1
8.	Singular Value Decomposition - Matrix Approximation	1
	Module-II (VECTOR CALCULUS)	6
1	Differentiation of Univariate Functions, Partial Differentiation and Gradients	1
2	Gradients of Vector Valued Functions, Gradients of Matrices	1
3	Useful Identities for Computing Gradients	1
4	Backpropagation and Automatic Differentiation	1
5	Higher Order Derivatives	1
6	Linearization and Multivariate Taylor Series	1
3	Module-III (Probability and Distributions)	10
1	Construction of a Probability Space - Discrete and Continuous Probabilities (Lecture 1)	1

2	Construction of a Probability Space - Discrete and Continuous Probabilities (Lecture 2)	1
3	Sum Rule, Product Rule	1
4	Bayes' Theorem	1
5	Summary Statistics and Independence	1
6	Important probability Distributions (Lecture 1)	1
7	Important probability Distributions (Lecture 2)	1
8	Conjugacy and the Exponential Family (Lecture 1)	1
9	Conjugacy and the Exponential Family (Lecture 2)	1
10	Change of Variables/Inverse Transform	1
4	Module-IV (Optimization)	7
1	Optimization Using Gradient Descent.	1
2	Gradient Descent With Momentum, Stochastic Gradient Descent	1
3	Constrained Optimization and Lagrange Multipliers (Lecture 1)	1
4	Constrained Optimization and Lagrange Multipliers (Lecture 2)	1
5	Convex Optimization	1
6.	Linear Programming	1
7.	Quadratic Programming	1
5	Module-V (CENTRAL MACHINE LEARNING PROBLEMS)	14
1.	Data and Learning models - Empirical Risk Minimization,	1
2.	Parameter Estimation	1
3.	Directed Graphical Models	1
4.	Linear Regression	1
5.	Bayesian Linear Regression	1
6.	Maximum Likelihood as Orthogonal Projection	1
7.	Dimensionality Reduction with Principal Component Analysis - Maximum Variance Perspective, Projection Perspective.	1
8.	Eigenvector Computation and Low Rank Approximations	1
9.	Density Estimation with Gaussian Mixture Models	1

10.	Parameter Learning via Maximum Likelihood	1
11.	EM Algorithm	1
12.	Classification with Support Vector Machines - Separating Hyperplanes	1
13.	Primal Support Vector Machines, Dual Support Vector Machines	1
14.	Kernels	1
*Assignments may include applications of the above theory. With respect to module V, programming assignments may be given.		



CST 285	DATA COMMUNICATION	Category	L	T	P	Credit	Year of Introduction
		MINOR	3	1	0	4	2019

Preamble: This is a basic course in communication for awarding B. Tech. Minor in Computer Science and Engineering with specialization in *Networking*. The purpose of this course is to prepare learners to understand the communication entities and the associated issues in the field of Computer Science. This course covers fundamental concepts of data transmission & media, digital & analog transmissions, multiplexing & spread spectrum, error detection & correction and switching. Concepts in data communication help the learner to understand the concepts in networking and mobile communication.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO1	Describe the characteristics of signals used for Analog and Digital transmissions (Cognitive knowledge: Understand)
CO2	Discuss the features and issues in data transmission (Cognitive knowledge: Understand)
CO3	Select transmission media based on characteristics and propagation modes (Cognitive knowledge: Apply)
CO4	Use appropriate signal encoding techniques for a given scenario (Cognitive knowledge: Apply)
CO5	Illustrate multiplexing and spread spectrum technologies (Cognitive knowledge: Understand)
CO6	Explain error detection & correction techniques and switching techniques used in data communication (Cognitive knowledge: Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓								✓		✓
CO2	✓	✓								✓		✓
CO3	✓											✓
CO4	✓	✓	✓	✓								✓
CO5	✓	✓	✓	✓						✓		✓
CO6	✓	✓	✓	✓						✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (Marks in percentage)	Test 2 (Marks in percentage)	End Semester Examination (Marks in percentage)
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Test : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus**Module 1****Data Transmission Basics**

Communication model - Simplex, Half duplex, Full duplex transmission. Periodic Analog signals - Sine wave, Amplitude, Phase, Wavelength, Time and frequency domain, Bandwidth. Analog & digital data and signals. Transmission impairments - Attenuation, Delay distortion, Noise. Data rate limits - Noiseless channel, Nyquist bandwidth, Noisy channel, Shannon's capacity formula.

Module 2**Transmission Media**

Guided Transmission Media - Twisted pair, Coaxial cable, Optical fiber. Unguided media - Radio waves, Terrestrial microwave, Satellite microwave, Infrared. Wireless Propagation - Ground wave propagation, Sky Wave propagation, Line-of-Sight (LoS) Propagation.

Module 3**Digital Transmission and Analog Transmission**

Digital data to Digital signal – Non-Return-to-Zero (NRZ), Return-to-Zero (RZ), Multilevel

binary, Biphase. Analog data to Digital signal - Sampling theorem, Pulse Code Modulation (PCM), Delta Modulation (DM). Digital data to Analog signal: Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), Phase Shift Keying (PSK). Analog data to Analog signal: Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM).

Module 4

Multiplexing and Spread Spectrum

Multiplexing - Frequency Division Multiplexing (FDM), Wave length Division Multiplexing (WDM), Time Division Multiplexing (TDM), Characteristics, Synchronous TDM, Statistical TDM. Spread Spectrum Techniques - Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), Code Division Multiplexing, Code Division Multiple Access (CDMA).

Module 5

Error Detection, Correction and Switching

Digital data communication techniques - Asynchronous transmission, Synchronous transmission. Detecting and correcting errors - Types of Errors, Parity check, Checksum, Cyclic Redundancy Check (CRC), Forward Error Correction (FEC), Hamming Distance, Hamming Code. Basic principles of Switching - Circuit Switching, Packet Switching, Message Switching.

Text Books

1. Forouzan B. A., Data Communications and Networking, 5/e, McGraw Hill, 2013.
2. William Stallings, Data and Computer Communication 9/e, Pearson Education, Inc.

Reference Books

1. Schiller J., Mobile Communications, 2/e, Pearson Education, 2009.
2. Curt M. White, Fundamentals of Networking and Communication 7/e, Cengage learning.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): What is a periodic analog signal? List the main properties of a periodic analog signal.

Course Outcome 2 (CO2): What is attenuation? How can it be handled?

Course Outcome 3 (CO3): How can interference be reduced using optical fiber?

Course Outcome 4 (CO4): Encode the data sequence 101011100 using Multilevel binary and Biphase schemes.

Course Outcome 5 (CO5): Explain direct sequence spread spectrum with a neat diagram.

Course Outcome 6 (CO6): Using Cyclic Redundancy Check (CRC), given the data-word 11110000 and the divisor 10011, show the generation of the codeword at the sender and the checking of the codeword at the receiver.

Model Question Paper

QP CODE:

PAGES: ____

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FOURTH SEMESTER B.TECH DEGREE (MINOR) EXAMINATION, MONTH & YEAR

Course Code: CST 285

Course name : DATA COMMUNICATION

Max Marks: 100

Duration: 3 Hours

PART-A

(Answer All Questions. Each question carries 3 marks)

1. What is bandwidth? Find the lowest frequency, if a periodic signal has a bandwidth of 20 Hz and the highest frequency is 60 Hz. Draw the Spectrum if the signal contains all frequencies of same amplitude.
2. Assume that a TV picture is to be transmitted over a channel with 4.5 MHz bandwidth and a 35 dB Signal-to-Noise-Ratio. Find the capacity of the channel.
3. What is the purpose of cladding in optical fibres?
4. Which wireless propagation is suitable for satellite communication? Justify your answer.
5. Explain the working of Delta Modulation with an example.
6. Illustrate the equivalent square wave pattern of the bit string 01001101 using Non-Return-to-Zero(NRZ) - Level and NRZ-Invert encoding schemes.
7. Distinguish between synchronous and statistical Time Division Multiplexing.
8. Apply Direct Sequence Spread Spectrum to the data 101 using the Barker sequence 10110111000. Show the encoding and decoding steps.
9. Find the minimum hamming distance for the following cases:
 - a) Detection of two errors
 - b) Correction of two errors
 - c) Detection of 3 errors or correction of 2 errors
 - d) Detection of 6 errors or correction of 2 errors
10. Find the parity bit for simple even parity check for the following.
 - a) 1001010
 - b) 0001100
 - c) 1000000
 - d) 1110111

PART-B

(Answer ANY one full question from each module. Each question carries 14 marks)

11. a) With the help of suitable figures, distinguish between time domain and frequency domain. (4)
- b) Describe the different types of transmission impairments. (10)

OR

12. a) Calculate the bandwidth, if a periodic signal is decomposed into 4 sine waves with frequencies 50 Hz, 100 Hz, 150 Hz and 200Hz. Draw the spectrum, assuming all components having amplitude in the range 6-12 V and all are multiple of two in the increasing order. (6)
- b) Distinguish between Nyquist bandwidth and Shannon capacity. Consider a noiseless channel with a bandwidth of 3000 Hz transmitting a signal with (i) Two signal levels and (ii) Four signal levels. Determine the maximum bit rate in both these cases. (8)
13. a) For a parabolic reflective antenna operating at 12 GHz with a diameter of 2 m, calculate the effective area and the antenna gain. (6)
- b) List any four advantages and disadvantages of twisted pair, coaxial cable and fiber optic cable. (8)

OR

14. a) Compare the features of terrestrial microwave and satellite microwave. (6)
- b) With the help of suitable diagrams, differentiate Multi-mode and Single-mode optical fibres. How the rays are propagated in Step-index and Graded-index Multi-mode fibres. (8)
15. a) Distinguish between data rate and signal rate. (4)

b) What is polar encoding? Encode the pattern 010011001110 using the two Biphasic schemes.

(10)

OR

16. a) Show the equivalent analog sine wave pattern of the bit string 010011010 using Amplitude Shift Keying, Frequency Shift Keying and Phase Shift Keying.

(4)

b) State Sampling theorem. Explain Pulse Code Modulation with suitable figures.

(10)

17. a) Four channels are multiplexed using Time Division Multiplexing. If each channel sends 100 bytes/sec and we multiplex one byte per channel, determine the frame size, duration of a frame, frame rate and bit rate of the link.

(6)

b) With the help of an example, explain the working of Frequency Hopping Spread Spectrum.

(8)

OR

18. a) Explain the different techniques by which the disparity in input data rate is handled by Time Division Multiplexing.

(4)

b) Suppose Alice and Bob are communicating using Code Division Multiple Access. Alice uses the code $[+1 +1]$ and Bob uses the code $[+1 -1]$. Alice sends the data bit 0 and Bob sends the data bit 1. Show the data in the channel and how they can detect what the other person has sent.

(10)

19. a) Explain parity check with examples.

(4)

b) Describe the need for a switch. What are the different phases in circuit switching?

(10)

OR

20. a) With the help of a suitable example, explain the virtual circuit approach of packet switching.

(6)

b) Find the Hamming code for the data-word 1011001. Assume odd parity.

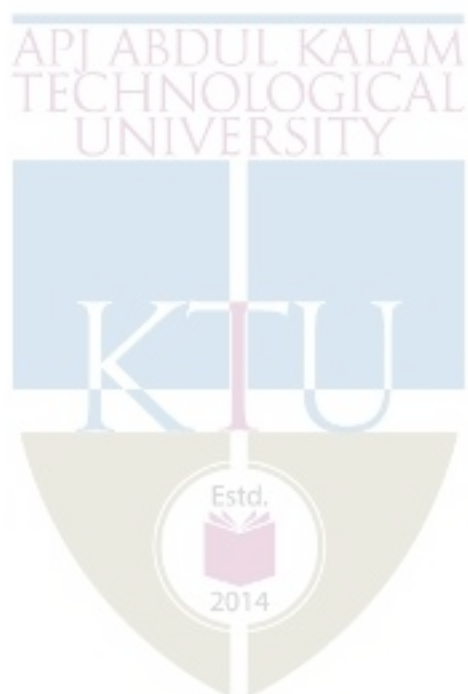
(8)

Teaching Plan

Module 1 : Data Transmission Basics		(8 Hours)
1.1	Introduction, Communication model - Simplex, Half duplex, Full duplex transmission	1
1.2	Periodic Analog signals - Sine wave, Amplitude, Phase, Wavelength	1
1.3	Time and frequency domain, Bandwidth	1
1.4	Analog data and signals	1
1.5	Digital data and signals	1
1.6	Transmission impairments - Attenuation, Delay distortion, Noise	1
1.7	Data rate limits - Noiseless channel, Nyquist bandwidth	1
1.8	Noisy channel, Shannon's capacity formula	1
Module 2: Transmission media		(7 Hours)
2.1	Guided Transmission Media - Twisted pair, Coaxial cable	1
2.2	Optical fiber	1
2.3	Unguided media - Radio waves	1
2.4	Terrestrial microwave, Satellite microwave	1
2.5	Infrared	1
2.6	Wireless Propagation - Ground wave propagation	1
2.7	Wave propagation, Line-of-Sight (LoS) Propagation	1
Module 3: Digital Transmission and Analog Transmission		(10 Hours)
3.1	Digital data to Digital signal – Non-Return-to-Zero (NRZ)	1
3.2	Return-to-Zero (RZ), Multilevel binary	1

3.3	Biphase	1
3.4	Analog data to Digital signal - Sampling theorem	1
3.5	Pulse Code Modulation (PCM)	1
3.6	Delta Modulation (DM)	1
3.7	Digital data to Analog signal: Amplitude Shift Keying (ASK)	1
3.8	Frequency Shift Keying (FSK), Phase Shift Keying (PSK)	1
3.9	Analog data to Analog signal: Amplitude Modulation (AM)	1
3.10	Frequency Modulation (FM), Phase Modulation (PM)	1
Module 4: Multiplexing and Spread Spectrum		(9 Hours)
4.1	Multiplexing - Frequency Division Multiplexing (FDM)	1
4.2	Wave length Division Multiplexing (WDM), Time Division Multiplexing (TDM)	1
4.3	Synchronous TDM, Statistical TDM	1
4.4	Spread Spectrum Techniques	1
4.5	Direct Sequence Spread Spectrum (DSSS)	1
4.6	Frequency Hopping Spread Spectrum (FHSS)	1
4.7	Code Division Multiplexing	1
4.8	Code Division Multiple Access (CDMA)	1
4.9	CDMA	1
Module 5: Error Detection, Correction and Switching		(11 Hours)
5.1	Digital data communication techniques - Asynchronous & Synchronous transmission	1
5.2	Detecting and correcting errors - Types of Errors	1
5.3	Parity check, Checksum	1
5.4	Cyclic Redundancy Check (CRC)	1
5.5	CRC	1
5.6	Forward Error Correction (FEC)	1
5.7	Hamming Distance, Hamming Code	1
5.8	Hamming Code	1
5.9	Basic principles of Switching - Circuit Switching	1

5.10	Packet Switching	1
5.11	Message Switching	1



CST 286	INTRODUCTION TO COMPUTER NETWORKS	Category	L	T	P	Credit	Year of Introduction
		MINOR	3	1	0	4	2019

Preamble: This is the second course for awarding B. Tech. Minor in Computer Science and Engineering with specialization in *Networking*. Study of this course provides the learners a clear understanding of how computer networks from local area networks to the massive and global Internet are built and how they allow the usage of computers to share information and communicate with one another. This course covers the layers of OSI Reference models and inter-networking. This course helps the learners to compare and analyze the existing network technologies and to choose a suitable network design for a given system.

Prerequisite: Data Communication (CST 255)

Course Outcomes: After the completion of the course the student will be able to

CO 1	Explain the features of computer networks, protocols and network design models (Cognitive Knowledge : Understand)
CO 2	Discuss the design issues of data link layer, data link layer protocols, bridges and switches (Cognitive Knowledge : Understand)
CO 3	Illustrate wired LAN protocols (IEEE 802.3/4/5) and wireless LAN protocols (IEEE 802.11a/b/g/n, 802.15) (Cognitive Knowledge : Understand)
CO 4	Select appropriate routing algorithms, congestion control techniques and Quality of Service requirements for a network (Cognitive Knowledge : Apply)
CO 5	Illustrate the functions and protocols of network layer, transport layer and application layer in inter-networking (Cognitive Knowledge : Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓								✓		✓
CO2	✓	✓	✓							✓		✓
CO3	✓	✓	✓							✓		✓
CO4	✓	✓	✓									✓
CO5	✓	✓	✓			✓				✓		✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Test 1 (Marks in percentage)	Test 2 (Marks in percentage)	End Semester Examination (Marks in percentage)
Remember	40	30	30
Understand	60	50	50
Apply		20	20
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Test : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus**Module 1**

Introduction – Uses of Computer Networks, Network Hardware, Network Software, Reference Models – The OSI Reference Model, The TCP/IP Reference Model, Comparison of OSI and TCP/IP Reference models.

Module 2

The Data Link Layer - Data Link layer Design Issues, Error Detection and Correction, Elementary Data Link Protocols, Sliding Window Protocols, HDLC (High-Level Data Link Control) Protocol. The Medium Access Control (MAC) Sub layer – The Channel Allocation Problem, Multiple Access Protocols, Ethernet, Wireless LANs - 802.11 a/b/g/n, Bridges & Switches.

Module 3

Network Layer Design Issues. Routing Algorithms - The Optimality Principle, Shortest path routing, Flooding, Distance Vector Routing, Link State Routing, Multicast Routing, Routing for Mobile Hosts. Congestion Control Algorithms, Quality of Service (QoS) - Requirements, Techniques for Achieving Good QoS.

Module 4

Network Layer in Internet – The IP Protocol, IP Addresses, Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (**RARP**), Bootstrap Protocol (**BOOTP**), Dynamic Host Configuration Protocol (DHCP). Open Shortest Path First (**OSPF**) Protocol, Border Gateway Protocol (**BGP**), Internet Multicasting, IPv6, ICMPv6.

Module 5

Transport Layer – The Transport Service – Services Provided to the Upper Layers, Transport Service Primitives. The User Datagram Protocol (UDP), Transmission Control Protocol (TCP) – Overview of TCP, TCP Segment Header, Connection Establishment & Release, Connection Management Modeling, TCP Retransmission Policy, TCP Congestion Control.

Application Layer – File Transfer Protocol (FTP), Domain Name System (DNS), Electronic mail, MIME, Simple Network Management Protocol (SNMP), World Wide Web – Architectural Overview.

Text Book

Andrew S. Tanenbaum, Computer Networks, 4/e, PHI (Prentice Hall India).

Reference Books

1. Behrouz A Forouzan, Data Communication and Networking, 4/e, Tata McGraw Hill
2. Larry L Peterson and Bruce S Dave, Computer Networks – A Systems Approach, 5/e, Morgan Kaufmann.
3. Fred Halsall, Computer Networking and the Internet, 5/e.
4. James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach, 6/e.
5. Keshav, An Engineering Approach to Computer Networks, Addison Wesley, 1998.
6. W. Richard Stevens. TCP/IP Illustrated volume 1, Addison-Wesley, 2005.
7. William Stallings, Computer Networking with Internet Protocols, Prentice-Hall, 2004.
8. Request for Comments (RFC) Pages - IETF -<https://www.ietf.org/rfc.html>

Sample Course Level Assessment Questions

CourseOutcome1 (CO1): Compare TCP/IP Reference model and OSI Reference model.

CourseOutcome2 (CO2): Distinguish between switches and bridges.

CourseOutcome3 (CO3): Draw and explain the frame format for Ethernet.

CourseOutcome5 (CO4): Discuss remedies for count to infinity problem in routing.

CourseOutcome4 (CO5): Subnet the Class C IP Address 206.16.2.0 so that you have 30 subnets. What is the subnet mask for the maximum number of hosts? How many hosts can each subnet have?

Model Question Paper

QP CODE: _____

PAGES: ____

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FOURTH SEMESTER B.TECH DEGREE (MINOR) EXAMINATION, MONTH & YEAR

Course Code: CST 286

Course name : INTRODUCTION TO COMPUTER NETWORKS

Max Marks: 100

Duration: 3 Hours

PART-A

(Answer All Questions. Each question carries 3 marks)

1. Why Layered Architecture is used in Computer Networks? Define the terms protocol and interface?
2. What are the different service primitives in Computer Networks?
3. Draw and explain Ethernet frame format.
4. What is the output string when the bit string 011110111110111110 is subjected to bit stuffing?
5. Discuss the count to infinity problem in routing.
6. What is flooding? Describe any two situations where flooding is advantageous.
7. What is IP (Internet Protocol) subnetting? Illustrate with example.
8. How many octets does the smallest possible IPv6 (IP version 6) datagram contain?
9. Can TCP (Transmission Control Protocol) be used directly over a network (e.g. an Ethernet) without using IP? Justify your answer
10. What is the role of SNMP (Simple Network Management Protocol)?

(10x3=30)

Part B

(Answer any one Question from each module. Each question carries 14 Marks)

Module I

11. (a) With a neat diagram, explain the OSI (Open Systems Interconnection) reference Model. (8)
(b) Compare OSI Reference model and the TCP/IP model (6)

OR

12. (a) Consider two networks providing reliable connection-oriented service. One of them offers a reliable byte stream and the other offers a reliable message stream. Are they identical? Justify your answer. (8)
(b) Compare LAN (Local Area Networks), MAN (Metropolitan Area Networks) and WAN (Wide Area Networks). (6)

Module II

13. (a) Discuss the different strategies used to avoid collisions in CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). (8)
(b) Briefly explain the working of HDLC (High-Level Data Link Control). (6)

OR

14. (a) Explain the working of IEEE 802.11. (10)
(b) Distinguish between Bridges and Switches. (4)

Module III

15. (a) Illustrate Distance Vector Routing Algorithm with an example. (8)
(b) Explain the characteristics of RIP (Routing Information Protocol). (6)

OR

16. (a) Explain an Interior Gateway protocol that uses a link state algorithm to propagate routing information. (6)
(b) Explain how routing is performed in a Mobile network. (8)

Module IV

17. (a) Explain address resolution problem and RARP (Reverse Address Resolution Protocol) with an example network. (10)
- (b) How IGMP (Internet Group Management Protocol) supports internet multicasting? Explain. (4)

OR

18. (a) Subnet the class C IP address 195.1.1.0 so that you have 10 subnets with a maximum of 12 hosts in each subnet. (6)
- (b) Draw IPv6 Datagram format and explain its features (8)

Module V

19. (a) Distinguish between TCP and UDP (User Datagram Protocol) header formats. (8)
- (b) Explain the principal DNS (Domain Name System) resource record types for IPv4. (6)

OR

20. (a) What is the role of SMTP (Simple Mail Transfer Protocol) in E-mail? (6)
- (b) With the help of a basic model explain the working of WWW (World Wide Web). (8)

Teaching Plan

Module 1		(8 Hours)
1.1	Introduction – Uses of Computer Networks.	1
1.2	Network Hardware – Local Area Networks (LAN), Metropolitan Area Networks (MAN), Wide Area Networks (WAN).	1
1.3	Network Hardware – Wireless Networks, Home Networks, Internetworks	1
1.4	Network Software — Protocol Hierarchies.	1
1.5	Network Software — Design issues for the layers.	1
1.6	Network Software – Connection Oriented and Connectionless Services, Service Primitives, Relationship of Services to Protocols.	1
1.7	Reference Models – The OSI Reference Model	1

1.8	Reference Models – The TCP/IP Reference Model, Comparison of OSI and TCP/IP Reference models	1
Module 2		(11 Hours)
2.1	Data Link layer Design Issues.	1
2.2	Error Detection and Correction - Error Correcting Codes	1
2.3	Error Detection and Correction - Error Detecting Codes	1
2.4	Elementary Data link Protocols.	1
2.5	Sliding Window Protocols.	1
2.6	HDLN (High-Level Data Link Control) Protocol	1
2.7	The Medium Access Control (MAC) Sub layer – The Channel Allocation Problem, Multiple Access Protocols.	1
2.8	Ethernet - Ethernet Cabling, Manchester Encoding, The Ethernet MAC Sub layer Protocol, The Binary Exponential Backoff Algorithm.	1
2.9	Ethernet - Ethernet Performance, Switched Ethernet, Fast Ethernet, Gigabit Ethernet, IEEE 802.2: Logical Link Control.	1
2.10	Wireless LANs - 802.11 a/b/g/n.	1
2.11	Bridges & Switches.	1
Module 3		(9 Hours)
3.1	Network Layer Design Issues.	1
3.2	Routing Algorithms - The Optimality Principle, Shortest path routing, Flooding.	1
3.3	Distance Vector Routing, Link State Routing.	1
3.4	Link State Routing.	1
3.5	Multicast Routing, Routing for Mobile Hosts	1
3.6	Distance Vector Routing, Link State Routing	1

3.7	Congestion control algorithms - General Principles of Congestion Control, Congestion Prevention Policies, Congestion Control in Virtual-Circuit Subnets	1
3.8	Congestion control algorithms - Congestion Control in Datagram Subnets, Load Shedding, Jitter Control	1
3.9	Quality of Service – Requirements, Techniques for Achieving Good Quality of Service.	1
Module 4		(9 Hours)
4.1	Network layer in internet, IP Protocol	1
4.2	IP Addresses – Subnets, Classless Inter Domain Routing (CIDR)	1
4.3	IP Addresses - Network Address Translation (NAT)	1
4.4	Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP),	1
4.5	Bootstrap Protocol (BOOTP), Dynamic Host Configuration Protocol (DHCP)	1
4.6	Open Shortest Path First (OSPF) Protocol	1
4.7	Border Gateway Protocol (BGP)	1
4.8	Internet Multicasting.	1
4.9	IPv6, Internet Control Message Protocol version 6 (ICMPv6).	1
Module 5		(8 Hours)
5.1	The Transport Service – Services Provided to the Upper Layers, Transport Service Primitives. The User Datagram Protocol (UDP)	1
5.2	Transmission Control Protocol (TCP) – Overview of TCP, TCP Segment Header, Connection Establishment & Release, Connection Management Modeling.	1
5.3	TCP Retransmission Policy, TCP Congestion Control.	1
5.4	Application Layer – File Transfer Protocol (FTP).	1
5.5	Domain Name System (DNS).	1

5.6	Electronic Mail.	1
5.7	Simple Network Management Protocol (SNMP)	1
5.8	World Wide Web – Architectural Overview	1



CST 381	CONCEPTS IN SOFTWARE ENGINEERING	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble: This course provides fundamental knowledge in the Software Development Process. It covers Software Development, Quality Assurance and Project Management concepts. This course enables the learners to apply state of the art industry practices in Software development.

Prerequisite: Basic understanding of Object Oriented Design and Development.

Course Outcomes: After the completion of the course the student will be able to

CO1	Differentiate Traditional and Agile Software Development approaches (Cognitive Knowledge Level: Understand)
CO2	Prepare Software Requirement Specification and Software Design for a given problem. (Cognitive Knowledge Level: Apply)
CO3	Justify the significance of design patterns and licensing terms in software development, prepare testing, maintenance and DevOps strategies for a project. (Cognitive Knowledge Level: Apply)
CO4	Make use of software project management concepts while planning, estimation, scheduling, tracking and change management of a project, with proper application of SCRUM, Kanban and Lean frameworks. (Cognitive Knowledge Level: Apply)
CO5	Utilize SQA practices, Process Improvement techniques and Technology improvements namely cloud based software model and containers & microservices in a Software Development Process. (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓						✓
CO2	✓	✓	✓	✓		✓				✓	✓	✓

CO3	✓	✓	✓	✓				✓		✓	✓	✓
CO4	✓	✓	✓	✓		✓			✓	✓	✓	✓
CO5	✓	✓	✓	✓		✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	30	30	30
Understand	30	30	30

Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks (Each student shall identify a software development problem and prepare Requirements Specification, Design Document, Project Plan and Test case documents for the identified problem as the assignment.)

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks.

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have a maximum of 2 subdivisions and carries 14 marks.

Syllabus**Module 1 : Introduction to Software Engineering (8 hours)**

Introduction to Software Engineering - Professional software development, Software engineering ethics. Software process models - The waterfall model, Incremental development. Process activities - Software specification, Software design and implementation, Software validation, Software evolution. Coping with change - Prototyping, Incremental delivery, Boehm's Spiral Model. Agile software development - Agile methods, agile manifesto - values and principles. Agile development techniques, Agile Project Management. Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care.

Module 2 : Requirement Analysis and Design (10 hours)

Functional and non-functional requirements, Requirements engineering processes. Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix. Developing use cases, Software Requirements Specification Template, Personas, Scenarios, User stories, Feature identification. Design concepts - Design within the context of software engineering, Design Process, Design concepts, Design Model. Architectural Design - Software Architecture, Architectural Styles, Architectural considerations, Architectural Design Component level design - What is a component?, Designing Class-Based Components, Conducting Component level design, Component level design for web-apps. Template of a Design Document as per “IEEE Std 1016-2009 IEEE Standard for Information Technology Systems Design Software Design Descriptions”. Case study: The Ariane 5 launcher failure.

Module 3 : Implementation and Testing (12 hours)

Object-oriented design using the UML, Design patterns, Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD. Review Techniques - Cost impact of Software Defects, Code review and statistical analysis. Informal Review, Formal Technical Reviews, Post-mortem evaluations. Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing, Debugging, White box testing, Path testing, Control Structure testing, Black box testing, Testing Documentation and Help facilities. Test automation, Test-driven development, Security testing. Overview of DevOps and Code Management - Code management, DevOps automation, CI/CD/CD. Software Evolution - Evolution processes, Software maintenance.

Module 4 : Software Project Management (8 hours)

Software Project Management - Risk management, Managing people, Teamwork. Project Planning, Software pricing, Plan-driven development, Project scheduling, Agile planning. Estimation techniques, COCOMO cost modeling. Configuration management, Version management, System building, Change management, Release management, Agile software management - SCRUM framework. Kanban methodology and lean approaches.

Module 5 : Software Quality and Process Improvement (6 hours)

Software Quality, Software Quality Dilemma, Achieving Software Quality Elements of Software Quality Assurance, SQA Tasks, Software measurement and metrics. Software Process Improvement(SPI), SPI Process CMMI process improvement framework, ISO 9001:2000 for Software.

Text Books

1. Book 1 - Ian Sommerville, Software Engineering, Pearson Education, Tenth edition, 2015.
2. Book 2 - Roger S. Pressman, Software Engineering : A practitioner's approach, McGraw Hill publication, Eighth edition, 2014
3. Book 3 - Ian Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering, Pearson Education, First Edition, 2020.

References

1. IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
2. IEEE Std 1016-2009 IEEE Standard for Information Technology—Systems Design—Software Design Descriptions
3. David J. Anderson, Kanban, Blue Hole Press 2010
4. David J. Anderson, Agile Management for Software Engineering, Pearson, 2003
5. Walker Royce, Software Project Management : A unified framework, Pearson Education, 1998
6. Steve. Denning, The age of agile, how smart companies are transforming the way work gets done. New York, Amacom, 2018.
7. Satya Nadella, Hit Refresh: The Quest to Rediscover Microsoft's Soul and Imagine a Better Future for Everyone, Harper Business, 2017
8. Henrico Dolfing, Project Failure Case Studies: Lessons learned from other people's mistakes, Kindle edition
9. Mary Poppendieck, Implementing Lean Software Development: From Concept to Cash, Addison-Wesley Signature Series, 2006
10. StarUML documentation - <https://docs.staruml.io/>
11. OpenProject documentation - <https://docs.openproject.org/>

12. BugZilla documentation - <https://www.bugzilla.org/docs/>
13. GitHub documentation - <https://guides.github.com/>
14. Jira documentation - <https://www.atlassian.com/software/jira>

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. What are the advantages of an incremental development model over a waterfall model?
2. Compare agile software development with traditional software development?

Course Outcome 2 (CO2):

1. How to prepare a software requirement specification?
2. Differentiate between Architectural design and Component level design.
3. How do agile approaches help software developers to capture and define the user requirements effectively?
4. What is the relevance of the SRS specification in software development?
5. Prepare a use case diagram for a library management system.

Course Outcome 3 (CO3):

1. Differentiate between the different types of software testing strategies.
2. What are the benefits of DevOps?
3. How do design patterns help software architects communicate the design of a complex system effectively?
4. What are the proactive approaches one can take to optimise efforts in the testing phase?

Course Outcome 4 (CO4):

1. What are the activities involved in software project management?
2. What is the need for SCRUM, Kanban and Lean methodologies?
3. What are the benefits of rolling level planning in software project management and how would you implement it?
4. How would you assess the risks in your software development project? How would you plan for risk mitigation and contingency?

Course Outcome 5 (CO5):

1. What is the importance of Software Process improvement?
2. How will retrospectives help in improving the software development process?
3. What are the important skills required for the SQA role?
4. How would you use project history data as a prediction tool to plan future projects?

Model Question Paper

QP CODE:

Reg No: _____

Name : _____

PAGES : 3

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FIFTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR**

Course Code: CST 381

Course Name: Concepts in Software Engineering

Duration: 3 Hrs

Max. Marks : 100

PART A

Answer all Questions. Each question carries 3 Marks

1. Explain why professional software that is developed for a customer is not simply the programs that have been developed and delivered
2. Incremental software development could be very effectively used for customers who do not have a clear idea about the systems needed for their operations. Discuss.
3. Identify and briefly describe four types of requirements that may be defined for a computer based system.
4. Describe software architecture in your own words.
5. What are the major differences between GPL and LGPL?
6. Compare between white box testing and black box testing.
7. What is the importance of risk management in software project management?
8. Explain COCOMO cost estimation model
9. Describe the software quality dilemma in your own words
10. Which are the levels of the CMMI model?

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 marks)

11. (a) Compare between waterfall model and spiral model (8)

(b) Explain Agile methods and Agile manifesto (6)

OR

12. (a) Explain software process activities (7)

(b) Explain Agile Development techniques and Agile Project Management. (7)

13. (a) What are functional and nonfunctional requirements? Imagine that you are developing a library management software for your college, identify at least 8 functional requirements and 4 nonfunctional requirements. (10)

(b) What are the contents of a software requirement specification? (4)

OR

14. (a) Explain Personas, Scenarios, User stories and Feature identification? (8)

(b) Compare between Software Architecture design and Component level design (6)

15. (a) Describe the formal and informal review techniques in detail. (6)

(b) Explain various software testing strategies. (8)

OR

16. (a) Explain DevOps CI/CD/CD in detail. (8)

(b) Explain test driven development. (6)

17. (a) What is a critical path and demonstrate its significance in a project schedule with the help of a sample project schedule. (6)

(b) Explain plan driven development and project scheduling (6)

OR

18. (a) Explain the SCRUM framework. (8)

(b) What is algorithmic cost modeling? What problems does it suffer from when compared with other approaches to cost estimation? (6)

19. (a) Explain elements of Software Quality Assurance and SQA Tasks. (8)

(b) Explain the SPI process. (6)

OR

20. (a) Compare between CMMI and ISO 9001:2000 (8)

(b) Compare Quality Control and Quality Assurance. (6)

Teaching Plan [44 hours]		
Module 1 : Introduction to Software Engineering (8 hours)		Hours
1.1	Introduction to Software Engineering. [Book 1, Chapter 1]	1 hour
1.2	Software process models [Book 1 - Chapter 2]	1 hour
1.3	Process activities [Book 1 - Chapter 2]	1 hour
1.4	Coping with change [Book 1 - Chapter 2, Book 2 - Chapter 4]	1 hour
1.5	Agile software development [Book 1 - Chapter 3]	1 hour
1.6	Agile development techniques [Book 1 - Chapter 3]	1 hour
1.7	Agile Project Management.[Book 1 - Chapter 3]	1 hour
1.8	Case studies : An insulin pump control system. Mentcare - a patient information system for mental health care. [Book 1 - Chapter 1]	1 hour
Module 2 : Requirement Analysis and Design (10 hours)		
2.1	Functional and non-functional requirements, Requirements engineering processes [Book 1 - Chapter 4]	1 hour

2.2	Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix [Book 1 - Chapter 4]	1 hour
2.3	Developing use cases, Software Requirements Specification Template [Book 2 - Chapter 8]	1 hour
2.4	Personas, Scenarios [Book 3 - Chapter 3]	1 hour
2.5	User stories, Feature identification [Book 3 - Chapter 3]	1 hour
2.6	Design concepts [Book 2 - Chapter 12]	1 hour
2.7	Architectural Design [Book 2 - Chapter 13]	1 hour
2.8	Component level design [Book 2 - Chapter 14]	1 hour
2.9	Component level design, Design Document Template. [Book 2 - Chapter 14, Ref - 2]	1 hour
2.10	Case study: The Ariane 5 launcher failure. [Book 2 - Chapter 16]	1 hour
Module 3 : Implementation and Testing (12 hours)		
3.1	Object-oriented design using the UML, Design patterns [Book 1 - Chapter 7]	1 hour
3.2	Implementation issues, Open-source development - Open-source licensing - GPL, LGPL, BSD [Book 1 - Chapter 7]	1 hour
3.3	Review Techniques - Cost impact of Software Defects, Code review. [Book 2 - Chapter 20]	1 hour
3.4	Informal Review, Formal Technical Reviews, Post-mortem evaluations. [Book 2 - Chapter 20]	1 hour
3.5	Software testing strategies [Book 2 - Chapter 22]	1 hour
3.6	Software testing strategies [Book 2 - Chapter 22]	1 hour
3.7	White box testing, Path testing, Control Structure testing [Book 2 - Chapter 23]	1 hour
3.8	Black box testing. Test documentation [Book 2 - Chapter 23]	1 hour
3.9	Test automation, Test-driven development [Book 3 - Chapter 9]	1 hour
3.10	Security testing. DevOps and Code Management [Book 3 - Chapter 9, Chapter 10]	1 hour
3.11	DevOps and Code Management - Code management, DevOps automation, CI/CD/CD. [Book 3 - Chapter 10]	1 hour

3.12	Software Evolution - Evolution processes, Software maintenance. [Book 1 - Chapter 9]	1 hour
Module 4 : Software Project Management (8 hours)		
4.1	Software Project Management - Risk management, Managing people, Teamwork [Book 1 - Chapter 22]	1 hour
4.2	Project Planning - Software pricing, Plan-driven development, Project scheduling, Agile planning [Book 1 - Chapter 23]	1 hour
4.3	Estimation techniques [Book 1 - Chapter 23]	1 hour
4.4	Configuration management [Book 1 - Chapter 25]	1 hour
4.5	Agile software management - SCRUM framework [Book 2 - Chapter 5]	1 hour
4.6	Agile software management - SCRUM framework [Book 2 - Chapter 5]	1 hour
4.7	Kanban methodology and lean approaches. [Ref 9 - Chapter 2]	1 hour
4.8	Kanban methodology and lean approaches.[Ref 9 - Chapter 2]	1 hour
Module 5 : Software Quality, Process Improvement and Technology trends (6 hours)		
5.1	Software Quality, Software Quality Dilemma, Achieving Software Quality. [Book 2 - Chapter 19]	1 hour
5.2	Elements of Software Quality Assurance, SQA Tasks [Book 3 - Chapter 21]	1 hour
5.3	Software measurement and metrics. [Book 3 - Chapter 21]	1 hour
5.4	Software Process Improvement(SPI), SPI Process[Book 2 - Chapter 37]	1 hour
5.5	Software Process Improvement(SPI), SPI Process[Book 2 - Chapter 37]	1 hour
5.6	CMMI process improvement framework, ISO 9001:2000 for Software. [Book 2 - Chapter 37]	1 hour

CST 382	INTRODUCTION TO SOFTWARE TESTING	Category	L	T	P	Credits	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This is a course in theoretical computer science that includes test cases for white-box, black-box, and grey-box approaches. This course describes the various techniques for test case design used to test software artifacts, including requirements, design, and code. The course includes different techniques for test case design based on graphs, programming language syntaxes and inputs. The course also covers symbolic execution using PEX tool.

Course Outcomes: After the completion of the course the student will be able to:-

CO1	List a range of different software testing techniques and be able to apply specific unit testing method to the projects using Junit. (Cognitive Knowledge Level: Understand)
CO2	Explain mutation testing method for a given piece of code to identify hidden defects that can't be detected using other testing methods. (Cognitive Knowledge Level: Understand)
CO3	Explain graph coverage criteria in terms of control flow graph and data flow graph for a given program. (Cognitive Knowledge Level: Understand)
CO4	Demonstrate the importance of black-box approaches in terms of domain and functional testing. (Cognitive Knowledge Level: Understand)
CO5	Illustrate the use of PEX tool with symbolic execution. (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO 9	PO10	PO11	PO12
CO1												
CO2												

CO3												
CO4												
CO5												

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Marks)	Test 2 (Marks)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment - Test : 25 marks

Continuous Assessment - Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First series test shall be preferably conducted after completing the first half of the syllabus and the second series test shall be preferably conducted after completing remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly completed module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly completed module), each with 7 marks. Out of the 7 questions, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carries 14 marks.

Syllabus

Module - 1 (Introduction to Software Testing)

Some Popular Errors – Ariane 5, Therac 25, Intel Pentium Bug. What is Software testing? Why should it be tested? Software Quality, Role of Testing. Testing Process - Level 0 thinking, Level 1 thinking, Level 2 thinking, Level 3 thinking, Level 4 thinking. Software Testing Terminologies - Verification, Validation and Testing, Faults, Error and Bug, Test cases, Coverage Criteria. Types of Testing- Unit testing, integration testing, System testing, Acceptance testing, Beta testing, Functional testing, Stress testing, Performance testing, Usability testing and Regression testing. Testing Methods - Black Box testing, White Box testing, Grey Box testing.

Module - 2 (Unit Testing)

Concept of Unit testing. Static Unit testing. Dynamic Unit testing - Control Flow testing, Data Flow testing, Domain testing, Functional Program testing. Mutation testing - Mutation and Mutants, Mutation operators, Mutation score. Junit - Framework for Unit testing. Case Study - Mutation testing using Junit and Muclipse.

Module - 3 (Unit Testing - White Box Approaches)

Overview of Graph Coverage Criteria. Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage, Complete path coverage, Prime path coverage, Complete round trip coverage, Simple round trip coverage. Data Flow Criteria - du paths, du pairs. Subsumption Relationships among Graph Coverage Criteria. Graph Coverage for Source Code - Control flow graphs for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Example program – Statistics. Graph Coverage for Design Elements - Call graphs and classes, Class inheritance testing: Coverage criteria, Coverage criteria on inheritance graph, Data flow at the design level, Inter-procedural DU pairs, Coupling du-pairs example. Example - Quadratic Root. Case Study - Graph Based testing using JUnit Framework.

Module - 4 (Unit Testing - Black Box Approaches)

Domain Testing / Input Space Partitioning - Partitions of a set. Input domain modelling - Interface-based approach, Functionality-based approach. Identifying values. Multiple partitions of the inputdomain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage. TriTyp example. Functional Testing - Functional Testing Concepts of Howden. Functional testing - Important Steps. Types of Functional testing - Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables, Random Testing. Case Study - Black Box testing approaches using JUnit.

Module - 5 (Grey Box Testing Approaches)

Introduction to Grey Box testing - Why Grey Box testing, Gray Box Methodology, Advantages and Disadvantages. Techniques of Grey Box Testing - Matrix Testing, Regression Testing, Orthogonal Array Testing or OAT, Pattern Testing. An Introduction to PEX - Parameterized Unit Testing, The Testing Problem. Symbolic Execution – Example, Symbolic execution tree. PEX application Case Study – PEX.

Text Books

1. Paul Ammann and Jeff Offutt, Introduction to Software Testing.
2. Kshirasagar Naik and Priyadarshi Tripathy, Software Testing And Quality Assurance: Theory And Practice.

Reference Materials

1. <https://www.csc.ncsu.edu/academics/undergrad/honors/thesis/muclipsebinder.pdf> - Muclipse tutorial.
2. King, James C, “Symbolic Execution and Program Testing”, Association for Computing Machinery, July 1976.
- 3.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): Explain the following types of testing methods with examples.

- (i) Black-box testing.
- (ii) White-box testing.
- (iii) Grey-box testing.

Course Outcome 2 (CO2): Define 12 mutants for the following method *power()* using effective mutation operators. Try to use each mutation operator at least once. Approximately, how many mutants do you think there would be, if all mutants for *power()* were created?

```
public static int power (int left, int right)
```

```
{
```

```
/**/
```

```
// Raises Left to the power of Right
```

```
// precondition : Right >= 0
```

```
// postcondition: Returns Left**Right
```

```
/**/
```

```
    intrslt;
```

```
    rslt = Left;
```



```

if (Right == 0)
{
    rslt = 1;
}
else
{
    for (int i = 2; i <= Right; i++)
        rslt = rslt * Left;
}
return (rslt);
}

```

Course Outcome 3 (CO3): Draw the control flow graph and data flow graph of given piece of code.

```

public static double ReturnAverage(int value[],int AS, int MIN, int MAX){
/*

```

Function: ReturnAverage Computes the average of all those numbers in the input array in the positive range [MIN, MAX]. The maximum size of the array is AS. But, the array size could be smaller than AS in which case the end of input is represented by -999.

```

*/

```

```

int i, ti, tv, sum;

```

```

double av;

```

```

i = 0; ti = 0; tv = 0; sum = 0;

```

```

while (ti < AS && value[i] != -999) {

```

```

    ti++;

```

```

    if (value[i] >= MIN && value[i] <= MAX) {

```

```

        tv++;

```

```

        sum = sum + value[i];

```

```

    }

```

```

    i++;

```

```

}

```

```

if (tv > 0)

```

```

    av = (double)sum/tv;

```

```

else
av = (double) -999;
return (av);
}

```

Course Outcome 4 (CO4): Explain the following with examples.

1. Input domain modelling.
2. All Combinations Coverage (ACoC)
3. Each Choice Coverage (ECC)
4. Pair-wise Coverage
5. T-wise Coverage
6. Base Choice Coverage
7. Multiple Base Choices Coverage.

Course Outcome 5 (CO5): Draw the symbolic execution tree for the following program code and explain the symbolic execution of testme (α_1 , α_2).

```

1. int twice (int v) {
2.   return 2 * v;
3. }
4. void testme (int x, int y ) {
5.   z = twice ( y);
6.   if ( z == x ){
7.     if ( x > y + 10)
8.       ERROR;
9.   }
10. }
11. int main() {
12.   x = sym input();
13.   y = sym input();
14.   testme ( x , y);
15.   return(0);
16. }

```

Model Question Paper

QP CODE:

PAGES: 4

Reg No: _____

Name : _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 382

Course Name: Introduction to Software Testing

Max.Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Explain the differences between Validation and Verification.
2. Explain the differences between Fault, Error, and Bug?
3. Define Ground string, Mutation score, and Mutants.
4. What are the functions of Test driver and Test stubs in dynamic unit testing?
5. Define Node coverage, Edge coverage and Prime path coverage in a control flow graph.
6. What are du paths and du pairs in a data flow graph?
7. Explain the two approaches in input domain modelling.
8. Explain the difference between Equivalence Class Partitioning and Boundary Value Analysis.
9. Briefly explain three techniques of Grey box testing.
10. Explain the concept of symbolic execution with the help of a toy example.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Explain the following types of testing
(i) Black Box testing (ii) White Box testing (iii) Grey Box testing

(14)

(iv) Unit testing (v) Integration testing (vi) System testing (vii) Acceptance testing

OR

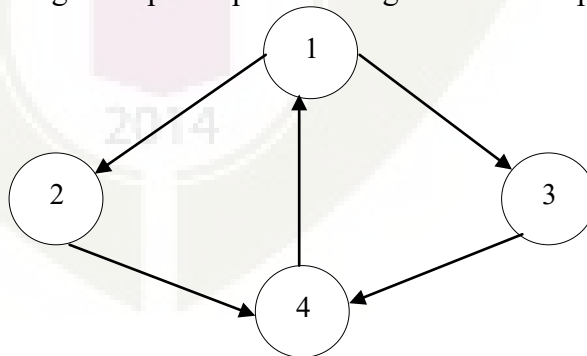
12. (a) Explain the following coverage criterias based on the code fragment given below. (i) Functional coverage (ii) Statement coverage (iii) Conditional coverage (iv) Branch coverage (8)

```
int foo (int x, int y){
    int z = 0;
    if ((x > 0) && (y > 0)){
        z = x;}
    return z;
}
```

- (b) Write positive and negative test cases for an ATM Machine? (6)
13. (a) Explain Dynamic unit test environment with a neat figure. (8)
- (b) Explain the major difference between control flow testing and data flow testing. (6)

OR

14. Explain seven types of mutation operators with neat examples. (14)
15. (a) Explain touring, side trips and detours with a neat example. (7)
- (b) Explain simple path coverage and prime path coverage with the help of CFG given below. (7)



OR

16. (a) Draw CFG fragment for

- (i) Simple *if* (ii) Simple *while* loop (iii) Simple *for* loop (7)
- (b) Explain the following concepts with examples. (7)
- (i) Call graph (ii) Inheritance graph (iii) Coupling du-pairs
17. (a) What are the four important steps in functional testing? (7)
- (b) Briefly explain input domain modelling approaches. (7)

OR

18. (a) Consider the triangle classification program with a specification: (6)
- The program reads floating values from the standard input. The three values A , B , and C are interpreted as representing the lengths of the sides of triangle. The program then prints a message to the standard output that states whether the triangle, if it can be formed, is scalene, isosceles, equilateral, or right angled. Determine the following for the above program:
- (i) For the boundary condition $A + B > C$ case (scalene triangle), identify test cases to verify the boundary.
- (ii) For the boundary condition $A = C$ case (isosceles triangle), identify test cases to verify the boundary.
- (iii) For the boundary condition $A = B = C$ case (equilateral triangle), identify test cases to verify the boundary.
- (b) Develop a decision table to generate test cases for this specification. (8)
19. (a) Explain the importance of grey box testing, its advantages and disadvantages. (9)
- (b) Explain the concept of symbolic execution tree. (5)

OR

20. (a) Consider the code fragment given below: - (7)
1. POWER: PROCEDURE(X , Y);
 2. $Z \leftarrow 1$;
 3. $J \leftarrow 1$;
 4. LAB: IF $Y \geq J$ THEN

```

5. DO; Z ← Z * X;
6. J ← J + 1;
7. GO TO LAB; END;
8. RETURN (Z) ;
9. END;

```

a) Explain Symbolic execution of POWER ($\alpha 1$, $\alpha 2$).

(b) Explain Execution tree for POWER ($\alpha 1$, $\alpha 2$) in the above code fragment. (7)

TEACHING PLAN

Index	Topics	No. of Hours (45)
Module 1 (Introduction to Software Testing) 9 Hours		
1.1	Some Popular Errors– Ariane 5, Therac 25, Intel Pentium Bug.	1 Hour
1.2	What is Software testing? Why should it be tested? Software Quality, Role of Testing.	1 Hour
1.3	Testing Process - Level 0 thinking, Level 1 thinking, Level 2 thinking, Level 3 thinking, Level 4 thinking.	1 Hour
1.4	Software Testing Terminologies- Verification, Validation and Testing, Faults, Error and Bug, Test cases, Coverage Criteria.	1 Hour
1.5	Types of Testing- Unit testing, integration testing, System testing, Acceptance testing, Beta testing	1 Hour
1.6	Functional testing, Stress testing	1 Hour
1.7	Performance testing, Usability testing and Regression testing.	1 Hour
1.8	Testing Methods - Black Box testing	1 Hour
1.9	Grey Box testing.	1 Hour
Module 2 (Unit testing) 8 Hours		

2.1	Concept of Unit testing.	1 Hour
2.2	Static Unit testing.	1 Hour
2.3	Dynamic Unit testing - Control Flow testing, Data Flow testing	1 Hour
2.4	Domain testing, Functional Program testing.	
2.5	Mutation testing - Mutation and Mutants, Mutation operators, Mutation score.	1 Hour
2.6	Junit - Framework for Unit testing.	1 Hour
2.7	Case Study - Mutation testing using Junit	1 Hour
2.8	Case Study - Mutation testing using Muclipse	1 Hour
Module 3 (Unit Testing:- White Box Approaches) 10 Hours		
3.1	Overview of Graph Coverage Criteria	1 Hour
3.2	Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage	1 Hour
3.3	Complete path coverage, Prime path coverage, Complete round trip coverage, Simple round trip coverage.	1 Hour
3.4	Data Flow Criteria - du paths, du pairs	1 Hour
3.5	Subsumption Relationships among Graph Coverage Criteria.	1 Hour
3.6	Graph Coverage for Source Code - Control Flow Graphs (CFG) for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Example program - Statistics	1 Hour
3.7	Graph Coverage for Design Elements - Call graphs and classes, Class inheritance testing: Coverage criteria, Coverage criteria on inheritance graph,	1 Hour

3.8	Data flow at the design level, Inter-procedural DU pairs, Coupling du-pairs example. Example - Quadratic Root	1 Hour
3.9	Case Study - Graph Based testing using JUnit Framework. (Lecture 1)	1 Hour
3.10	Case Study - Graph Based testing using JUnit Framework. (Lecture 2)	1 Hour
Module 4 (Unit Testing:- Black Box Approaches) 9 Hours		
4.1	Domain Testing / Input Space Partitioning - Partitions of a set.	1 Hour
4.2	Input domain modelling - Interface-based approach, Functionality-based approach.	1 Hour
4.3	Identifying values.	1 Hour
4.4	Multiple partitions of the input domain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage.	1 Hour
4.5	TriTyp example.	1 Hour
4.6	Functional Testing - Functional Testing Concepts of Howden. Important Steps.	1 Hour
4.7	Types of Functional testing - Equivalence Class Partitioning, Boundary Value Analysis	1 Hour
4.8	Decision Tables, Random Testing.	1 Hour
4.9	Case Study - Black Box testing approaches using JUnit.	1 Hour
Module 5 (Grey Box Testing Approaches) 9 Hours		
5.1	Introduction to Grey Box testing - Why Grey Box testing, Gray Box Methodology, Advantages and Disadvantages.	1 Hour
5.2	Techniques of Grey Box Testing - Matrix Testing, Regression Testing,	1 Hour

5.3	Orthogonal Array Testing or OAT, Pattern Testing.	1 Hour
5.4	An Introduction to Pex - Parameterized Unit Testing, The Testing Problem.	1 Hour
5.5	Symbolic Execution – Example, Symbolic execution tree.	1 Hour
5.6	PEX application.	1 hour
5.7	Case Study – PEX (Lecture 1)	1 Hour
5.8	Case Study – PEX (Lecture 2)	1 Hour
5.9	Case Study – PEX (Lecture 3)	1 Hour

CST 383	CONCEPTS IN MACHINE LEARNING	Category	L	T	P	Credit	Year of introduction
		VAC	3	1	0	4	2019

Preamble: This course enables the learners to understand the fundamental concepts and algorithms in machine learning. The course covers the standard and most popular supervised learning algorithms such as linear regression, logistic regression, decision trees, Bayesian learning & the naive Bayes algorithm, support vector machines & kernels, basic clustering algorithms and dimensionality reduction methods. This course helps the students to provide machine learning based solutions to real world problems.

Prerequisite: Familiarity with basics in linear algebra, probability and Python programming.

	Course Outcomes
CO1	Illustrate Machine Learning concepts and basic parameter estimation methods. (Cognitive Knowledge Level: Apply)
CO2	Demonstrate supervised learning concepts (regression, linear classification). (Cognitive Knowledge Level: Apply)
CO3	Illustrate the concepts of Multilayer neural network and Support Vector Machine (Cognitive Knowledge Level: Apply)
CO4	Describe unsupervised learning concepts and dimensionality reduction techniques. (Cognitive Knowledge Level: Apply)
CO5	Solve real life problems using appropriate machine learning models and evaluate the performance measures (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓	✓							✓
CO2	✓	✓	✓	✓	✓							✓
CO3	✓	✓	✓	✓	✓							✓

CO4	✓	✓	✓	✓	✓							✓
CO5	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	30	30	30
Understand	30	30	30
Apply	40	40	40
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks
 Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : **15 marks**

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks

First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus

Module-1 (Overview of machine learning)

Machine learning paradigms-supervised, semi-supervised, unsupervised, reinforcement learning. Basics of parameter estimation - maximum likelihood estimation(MLE) and maximum a posteriori estimation(MAP). Introduction to Bayesian formulation.

Module-2 (Supervised Learning)

Regression - Linear regression with one variable, Linear regression with multiple variables, solution using gradient descent algorithm and matrix method, basic idea of overfitting in regression. Linear Methods for Classification- Logistic regression, Perceptron, Naive Bayes, Decision tree algorithm ID3.

Module-3 (Neural Networks (NN) and Support Vector Machines (SVM))

NN - Multilayer feed forward network, Activation functions (Sigmoid, ReLU, Tanh), Backpropagation algorithm.

SVM - Introduction, Maximum Margin Classification, Mathematics behind Maximum Margin Classification, Maximum Margin linear separators, soft margin SVM classifier, non-linear SVM, Kernels for learning non-linear functions, polynomial kernel, Radial Basis Function(RBF).

Module-4 (Unsupervised Learning)

Clustering - Similarity measures, Hierarchical Agglomerative Clustering, K-means partitional clustering, Expectation maximization (EM) for soft clustering. Dimensionality reduction – Principal Component Analysis, factor Analysis, Multidimensional scaling, Linear Discriminant Analysis.

Module-5 (Classification Assessment)

Classification Performance measures - Precision, Recall, Accuracy, F-Measure, Receiver Operating Characteristic Curve(ROC), Area Under Curve(AUC. Bootstrapping, Cross Validation, Ensemble methods, Bias-Variance decomposition. Case Study: Develop a classifier for face detection.

Text Book

1. Ethem Alpaydin, Introduction to Machine Learning, 2nd edition, MIT Press 2010.
2. Mohammed J. Zaki and Wagner Meira, Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, First South Asia edition, 2016.
3. Jake VanderPlas, Python Data Science Handbook, O'Reilly Media, 2016
4. Tom Mitchell, Machine Learning, McGraw-Hill, 1997.

Reference Books

1. Christopher Bishop. Neural Networks for Pattern Recognition, Oxford University Press, 1995.
2. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective, MIT Press 2012.
3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements Of Statistical Learning, Second edition Springer 2007.
4. P. Langley, Elements of Machine Learning, Morgan Kaufmann, 1995.
5. Richert and Coelho, Building Machine Learning Systems with Python.
6. Davy Cielen, Arno DB Meysman and Mohamed Ali. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Dreamtech Press 2016.

Sample Course Level Assessment Questions

Course Outcome1(CO1):

1. A coin is tossed 100 times and lands heads 62 times. What is the maximum likelihood estimate for θ , the probability of heads.
2. Suppose data x_1, \dots, x_n are independent and identically distributed drawn from an exponential distribution $\exp(\lambda)$. Find the maximum likelihood for λ .
3. Suppose x_1, \dots, x_n are independent and identically distributed(iid) samples from a distribution with density

$$f_X(x|\theta) = \begin{cases} \frac{\theta x^{\theta-1}}{3^\theta}, & 0 \leq x \leq 3 \\ 0, & \text{otherwise} \end{cases}$$

Find the maximum likelihood estimate(MLE) for θ .

4. Find the maximum likelihood estimator (MLE) and maximum a posteriori (MAP) estimator for the mean of a univariate normal distribution. Assume that we have N samples, x_1, \dots, x_N independently drawn from a normal distribution with known variance σ^2 and unknown mean μ and the prior distribution for the mean is itself a normal distribution with mean ν and variance β^2 . What happens to the MLE and MAP estimators as the number of samples goes to infinity.

Course Outcome 2 (CO2):

1. Explain the difference between (batch) gradient descent and stochastic gradient descent. Give an example of when you might prefer one over the other.
2. Suppose that you are asked to perform linear regression to learn the function that outputs y , given the D -dimensional input x . You are given N independent data points, and that all the D attributes are linearly independent. Assuming that D is around 100, would you prefer the closed form solution or gradient descent to estimate the regressor?
3. Suppose you have a three class problem where class label $y \in 0, 1, 2$ and each training example X has 3 binary attributes $X_1, X_2, X_3 \in 0, 1$. How many parameters (probability distribution) do you need to know to classify an example using the Naive Bayes classifier?

Course Outcome 3 (CO3):

1. What are support vectors and list any three properties of the support vector classifier solution?
2. Why do you use kernels to model a projection from attributes into a feature space, instead of simply projecting the dataset directly?
3. Describe how Support Vector Machines can be extended to make use of kernels. Illustrate with reference to the Gaussian kernel $K(x, y) = e^{-z}$, where $z = (x-y)^2$.

4. Briefly explain one way in which using tanh instead of logistic activations makes optimization easier.
5. ReLU activation functions are most used in neural networks instead of the tanh activation function. Draw both activation functions and give a) an advantage of the ReLU function compared to the tanh function. b) a disadvantage of the ReLU function compared to the tanh function.

Course Outcome 4(CO4):

1. Describe cluster analysis? Identify two applications where cluster analysis can be applied to multimedia data?
2. Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):
 - (i) Compute the Euclidean distance between the two objects.
 - (ii) Compute the Manhattan distance between the two objects.
3. Use PCA to reduce the dimension from 2 to 1 for the design matrix X .

$$X = \begin{bmatrix} 6 & -4 \\ -3 & 5 \\ -2 & 6 \\ 7 & -3 \end{bmatrix}$$

4. What is Principal Component Analysis (PCA)? Which eigen value indicates the direction of largest variance?
5. Suppose that one runs a principal component analysis on a data set and tells that the percentage of variance explained by the first 3 components is 80%. How is this percentage of variance explained?

Course Outcome 5 (CO5):

1. Suppose that you are contacted by a food processing company that wants you to develop a classifier that detects whether a rat is present in an image. You collect a large dataset of images by crawling the web, and have annotators determine which images contain rats. This set of images can then be used as the training set for your classifier.
 - a. Suggest a machine learning method to use for this classification task and evaluate its performance.
 - b. After you have delivered your solution to the company, they get back to you and complain that when they evaluate on a new test set, they get precision and recall values that are much lower than what you reported to them. Explain what might have gone wrong and propose remedial measures .
2. A real estate firm would like to build a system that predicts the sale prices of a house. They create a spreadsheet containing information about 1,500 house sales in the Kochi

area. In addition to the price, there are 10 features describing the house, such as number of bedrooms, total indoor area, lot area, a swimming pool, location, etc. Explain how you would implement a machine learning model that would solve this prediction task. Give all steps you would carry out when developing it. Explain why the model you built is probably useless in the long run.

3. For a classifier, the confusion matrix is given by:

	+	-
+	9	9
-	1	5

What is the precision, recall and accuracy of that classifier?

Model Question Paper

QP CODE: _____

PAGES:3

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
FIFTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH &
YEAR

Course Code: CST 383

Course Name: CONCEPTS IN MACHINE LEARNING

Max.Marks:100

Duration: 3

Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Define supervised learning? Name special cases of supervised learning depending on whether the inputs/outputs are categorical, or continuous.
2. Differentiate between Maximum Likelihood estimation (MLE) and Maximum a Posteriori (MAP) estimation?
3. What is overfitting and why is it a problem? Give an example of a method to reduce the risk of overfitting.
4. Specify the basic principle of gradient descent algorithm.
5. Suppose that you have a linear support vector machine(SVM) binary classifier. Consider a point that is currently classified correctly, and is far away from the decision boundary. If you

remove the point from the training set, and re-train the classifier, will the decision boundary change or stay the same? Justify your answer.

6. Mention the primary motivation for using the kernel trick in machine learning algorithms?
7. Expectation maximization (EM) is designed to find a maximum likelihood setting of the parameters of a model when some of the data is missing. Does the algorithm converge? If so, do you obtain a locally or globally optimal set of parameters?
8. Illustrate the strength and weakness of k-means algorithm.
9. Classifier A attains 100% accuracy on the training set and 70% accuracy on the test set. Classifier B attains 70% accuracy on the training set and 75% accuracy on the test set. Which one is a better classifier. Justify your answer.
10. How does bias and variance trade-off affect machine learning algorithms?

(10x3=30)

Part B

Answer any one Question from each module. Each question carries 14 Marks

11. a) Suppose that X is a discrete random variable with the following probability mass function: where $0 \leq \theta \leq 1$ is a parameter. The following 10 independent observations

X	0	1	2	3
$P(X)$	$2\theta/3$	$\theta/3$	$2(1 - \theta)/3$	$(1 - \theta)/3$

were taken from such a distribution: (3, 0, 2, 1, 3, 2, 1, 0, 2, 1). What is the maximum likelihood estimate of θ . (6)

- b) A gamma distribution with parameters α, β has the following density function, where $\Gamma(t)$ is the gamma function.

$$p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$$

If the posterior distribution is in the same family as the prior distribution, then we say that the prior distribution is the conjugate prior for the likelihood function. Using the Gamma distribution as a prior, show that the Exponential distribution is a conjugate prior of the Gamma distribution. Also, find the maximum a posteriori estimator for the parameter of the Exponential distribution as a function of α and β . (8)

OR

12. a) Traffic between 8AM and 9AM at a certain place was measured by counting the number of vehicles that passed at that time. Suppose the counts follow a Poisson process. A random sample of 9 observations was collected, having observed the following number of vehicles: (95, 100, 80, 70, 110, 98, 97, 90, 70). Derive the maximum likelihood estimator for the

average number of vehicles that pass by that place between 8 AM and 9 AM, and compute the corresponding estimate using the given sample. (7)

- b) Find the maximum a posteriori (MAP) estimator for the mean of a univariate normal distribution. Assume that we have N samples, x_1, \dots, x_N independently drawn from a normal distribution with known variance σ^2 and unknown mean μ and the prior distribution for the mean is itself a normal distribution with mean ν and variance β^2 . (7)

- 13.a) Derive the gradient descent training rule assuming for the target function $o_d = w_0 + w_1 x_1 + \dots + w_n x_n$. Define explicitly the squared cost/error function E , assuming that a set of training examples D is provided, where each training example $d \in D$ is associated with the target output t_d . (10)

- b) How can we interpret the output of a two-class logistic regression classifier as a probability? (4)

OR

14. a) In a two-class logistic regression model, the weight vector $w = [4, 3, 2, 1, 0]$. We apply it to some object that we would like to classify; the vectorized feature representation of this object is $x = [-2, 0, -3, 0.5, 3]$. What is the probability, according to the model, that this instance belongs to the positive class? (6)

- b) The following dataset can be used to train a classifier that determines whether a given person is likely to own a car or not. There are three features: education level (primary, secondary, or university); residence (city or country); gender (female, male).

education	residence	gender	has car?
sec	country	female	yes
univ	country	female	yes
prim	city	male	no
univ	city	male	no
sec	city	female	no
sec	country	male	yes
prim	country	female	yes
univ	country	male	yes
sec	city	male	yes
prim	city	female	no
univ	city	female	no
prim	country	male	yes

Find the root attribute and justify your answer (8)

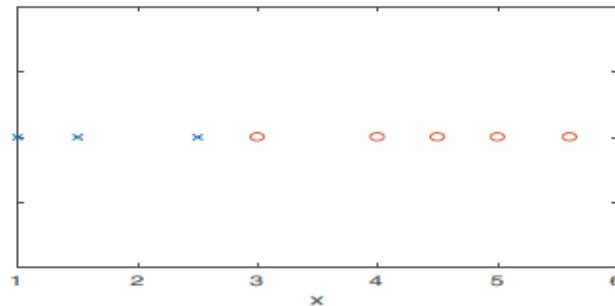
15. a) Consider a support vector machine whose input space is 2-D, and the inner products are computed by means of the kernel $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^2 - 1$, where $\mathbf{x} \cdot \mathbf{y}$ denotes the ordinary inner product. Show that the mapping to feature space that is implicitly defined by this kernel is the mapping to 5-D given by (10)

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2} x_1 x_2 \\ \sqrt{2} x_1 \\ \sqrt{2} x_2 \end{bmatrix}.$$

- b) What is the basic idea of a Support Vector Machine? (4)

OR

16. a) Explain how back propagation can be used to solve XOR problem which is not linearly separable. (8)
- b) Consider the following one dimensional training data set, 'x' denotes negative examples and 'o' positive examples. The exact data points and their labels are given in the table. Suppose a SVM is used to classify this data. Indicate which are the support vectors and mark the decision boundary. Find the equation of the hyperplane. (6)



x	1	1.5	2.5	3	4	4.5	5	5.6
y	-1	-1	-1	1	1	1	1	1

17. a) Suppose that we have the following data (one variable). Use single linkage Agglomerative clustering to identify the clusters.
Daa: (2, 5, 9, 15, 16, 18, 25, 33, 33, 45). (8)
- b) Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):
- Compute the Euclidean distance between the two objects.
 - Compute the Manhattan distance between the two objects.

- (iii) Compute the Minkowski distance between the two objects, using $p = 3$ (6)

OR

18. a) Suppose that we have the following data:

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>
(2,0)	(1,2)	(2,2)	(3,2)	(2,3)	(3,3)	(2,4)	(3,4)	(4,4)	(3,5)

Identify the cluster by applying the k-means algorithm, with $k = 2$. Try using initial cluster centers as far apart as possible. (10)

- b) List the steps involved in Principal Component Analysis. (4)

19. a) Suppose the dataset had 9700 cancer-free images from 10000 images from cancer patients. Find precision, recall and accuracy? Is it a good classifier? Justify. (8)

Actual Class\ Predicted class	cancer = yes	cancer = no	Total
cancer = yes	90	210	300
cancer = no	140	9560	9700
Total	230	9770	10000

- b) Suppose that you have a classification problem where our feature representation contains about 10,000,000 features. We would like to develop a classifier that can be deployed in a mobile phone, so preferably it should have a small memory footprint. Discuss one solution for how this can be done. (6)

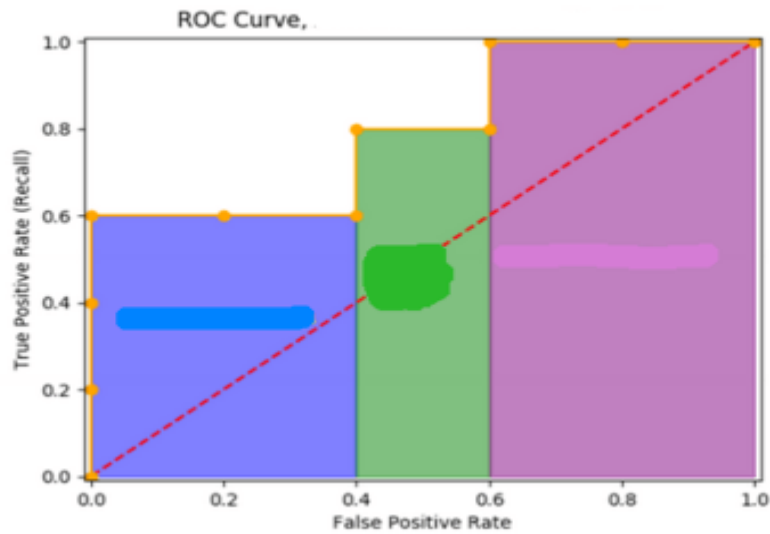
OR

20. a) What are ROC space and ROC curve in machine learning? In ROC space, which points correspond to perfect prediction, always positive prediction and always negative prediction? Why? (6)

- b) Suppose there are three classifiers A, B and C. The (FPR, TPR) measures of the three classifiers are as follows – A (0, 1), B (1, 1), C (1, 0.5). Which can be considered as a perfect classifier? Justify your answer. (4)

c) Given the following ROC Curve? Find the AUC?

(4)



Teaching Plan

No	Contents	No of Lecture Hrs
Module 1 :Overview of machine learning (7 hours)		
1.1	Supervised, semi-supervised, unsupervised learning, reinforcement learning (Text Book (TB) 1: Chapter 1)	1hour
1.2	Maximum likelihood estimation(MLE) (TB 1: Section 4.2)	1hour
1.3	Maximum likelihood estimation (MLE)- example (TB 1: Section 4.2)	1hour
1.4	Maximum a posteriori estimation(MAP) (TB 4: Section 6.2)	1hour
1.5	Maximum a posteriori estimation(MAP)-example (TB 4: Section 6.2)	1hour
1.6	Bayesian formulation (TB 1: Section 14.1, 14.2)	1hour
1.7	Bayesian formulation -example (TB 1: Section 14.1, 14.2)	1hour
Module 2 : Supervised Learning (8 hours)		

2.1	Linear regression with one variable (TB 1: Section 2.6)	1 hour
2.2	Multiple variables, Solution using gradient descent algorithm and matrix method (No derivation required) (TB 1: Section 5.8)	1 hour
2.3	Overfitting in regression, Lasso and Ridge regularization	1 hour
2.4	Logistic regression	1 hour
2.5	Perceptron	1 hour
2.6	Naive Bayes (TB 2: Section 18.2)	1 hour
2.7	Decision trees (TB 2: Chapter 19)	1 hour
2.8	Decision trees- ID3 algorithm (TB 2: Chapter 19)	1 hour
Module 3 : Neural Networks and Support Vector Machines (TB 2: Chapter 21) (11 hours)		
3.1	Multilayer Feed forward Network, Activation Functions (Sigmoid, ReLU, Tanh)	1 hour
3.2	Back Propagation Algorithm	1 hour
3.3	Illustrative Example for Back Propagation	1 hour
3.4	Introduction, Maximum Margin Hyperplane,	1 hour
3.5	Mathematics behind Maximum Margin Classification	1 hour
3.6	Formulation of maximum margin hyperplane and solution	1 hour
3.7	Soft margin SVM	1 hour
3.8	Solution of Soft margin SVM	1 hour
3.9	Non-linear SVM	1 hour
3.10	Kernels for learning non-linear functions and properties of kernel functions.	1 hour
3.11	Example Kernels functions- Linear, RBF, Polynomial.	1 hour
Module 4 : Unsupervised Learning (10 hours)		
4.1	Similarity measures- Minkowski distance measures(Manhattan, Euclidean), Cosine Similarity	1 hour
4.2	Clustering - Hierarchical Clustering (TB 2: Chapter 14)	1 hour
4.3	K-means partitional clustering (TB 2: Chapter 13)	1 hour
4.4	Expectation maximization (EM) for soft clustering (TB 2: Chapter 13)	1 hour
4.5	Expectation maximization (EM) for soft clustering (TB 2: Chapter 13)	1 hour

4.6	Dimensionality reduction – Principal Component Analysis (TB 1: Section 6.3)	1hour
4.7	Dimensionality reduction – Principal Component Analysis (TB 1: Section 6.3)	1hour
4.8	Factor Analysis (TB 1: Section 6.4)	1hour
4.9	Multidimensional scaling (TB 1: Section 6.5)	1hour
4.10	Linear Discriminant Analysis (TB 1: Section 6.6)	1hour
Module 5 : Classification Assessment (8 hours)		
5.1	Performance measures - Precision, Recall, Accuracy, F-Measure, ROC, AUC. (TB 2: Chapter 22.1)	1hour
5.2	Boot strapping, Cross validation	1hour
5.3	Ensemble methods- bagging	1hour
5.4	Ensemble methods- boosting	1hour
5.5	Bias-Variance decomposition (TB 2: Chapter 22.3)	1hour
5.6	Bias-Variance decomposition (TB 2: Chapter 22.3)	1hour
5.7	Face detection (TB 3: Chapter 5 Section Application: A Face Detection Pipeline)	1hour
5.8	Face detection (TB 3: Chapter 5 Section Application: A Face Detection Pipeline)	1hour



CST 384	CONCEPTS IN DEEP LEARNING	Category	L	T	P	Credits	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This course aims to introduce the learner to an overview of the concepts and algorithms involved in deep learning. Deep learning is a subfield of machine learning, a subfield of artificial intelligence. Basic concepts and application areas of machine learning, deep networks, convolutional neural network and recurrent neural network are covered here. This is a foundational program that will help students understand the capabilities, challenges, and consequences of deep learning and prepare them to participate in the development of leading-edge AI technology. They will be able to gain the knowledge needed to take a definitive step in the world of AI.

Prerequisite: Sound knowledge in Basics of linear algebra and probability theory.

CO1	Demonstrate basic concepts in machine learning.(Cognitive Knowledge Level: Understand)
CO2	Illustrate the validation process of machine learning models using hyper-parameters and validation sets. (Cognitive Knowledge Level: Understand)
CO3	Demonstrate the concept of the feed forward neural network and its training process. (Cognitive Knowledge Level: Apply)
CO4	Build CNN and Recurrent Neural Network (RNN) models for different use cases. (Cognitive Knowledge Level: Apply)
CO5	Use different neural network/deep learning models for practical applications. (Cognitive Knowledge Level: Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓								✓
CO2	✓	✓	✓	✓								✓
CO3	✓	✓	✓	✓	✓							✓
CO4	✓	✓	✓	✓	✓	✓						✓
CO5	✓	✓	✓	✓	✓	✓						✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test1 (Percentage)	Test2 (Percentage)	
Remember	30	30	30
Understand	40	40	40
Apply	30	30	30
Analyse			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Tests : 25 marks

Continuous Assessment Assignment : 15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First Internal Examination shall be preferably conducted after completing the first half of the syllabus and the Second Internal Examination shall be preferably conducted after completing remaining part of the syllabus.

There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Syllabus

INTRODUCTION TO DEEP LEARNING

(General Instructions: Instructors are to introduce students to any one software platform and demonstrate the working of the algorithms in the syllabus using suitable use cases and public datasets to give a better understanding of the concepts discussed. Tutorial hour may be used for this purpose)

Module-1 (Introduction)

Key components - Data, models, objective functions, optimization algorithms, Learning algorithm. Supervised learning- regression, classification, tagging, web search, page ranking, recommender systems, sequence learning, Unsupervised learning, Reinforcement learning, Historical Trends in Deep Learning. Other Concepts - overfitting, underfitting, hyperparameters and validation sets, estimators, bias and variance.

Module- 2 (Optimization and Neural Networks)

Neural Networks –Perceptron, Gradient Descent solution for Perceptron, Multilayer perceptron, activation functions, architecture design, chain rule, back propagation, gradient based learning. Introduction to optimization– Gradient based optimization, linear least squares. Stochastic gradient descent, Building ML algorithms and challenges.

Module -3 (Convolutional Neural Network)

Convolutional Neural Networks – convolution operation, motivation, pooling, Structure of CNN, Convolution and Pooling as an infinitely strong prior, variants of convolution functions, structured outputs, data types, efficient convolution algorithms. Practical challenges of common deep learning architectures- early stopping, parameter sharing, dropout. Case study: AlexNet, VGG, ResNet.

Module- 4 (Recurrent Neural Network)

Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU, Practical use cases for RNNs.

Module-5 (Application Areas)

Applications – computer vision, speech recognition, natural language processing, common word embedding: continuous Bag-of-Words, Word2Vec, global vectors for word representation (GloVe). Research Areas – autoencoders, representation learning, boltzmann machines, deep belief networks.

Text Book

1. Ian Goodfellow, YoshuaBengio, Aaron Courville, Deep Learning, MIT Press 2015 ed.
2. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Dive into Deep Learning, August 2019.
3. Neural Networks and Deep Learning, Aggarwal, Charu C., c Springer International Publishing AG, part of Springer Nature 2018

Reference Books

1. Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks by Russell Reed, Robert J MarksII, A Bradford Book,2014
2. Practical Convolutional Neural Networks by MohitSewak, Md. Rezaul Karim, PradeepPujari, Packt Publishing 2018
3. Hands-On Deep Learning Algorithms with Python by SudharsanRavichandran, Packt Publishing 2019
4. Deep Learning with Python by Francois Chollet, Manning Publications Co., 2018

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

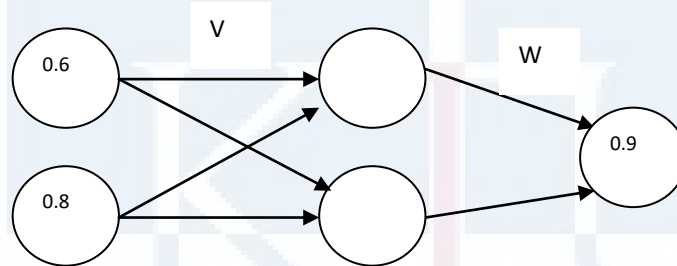
1. Compare regression and classification.
2. Define supervised learning? Distinguish between regression and classification.
3. Discuss the different learning approaches used in machine learning.

Course Outcome 2 (CO2):

1. What are hyperparameters? Why are they needed?
2. What issues are to be considered while selecting a model for applying machine learning in a given problem?

Course Outcome 3 (CO3):

1. Update the parameters V_{11} in the given MLP using back propagation with learning rate as 0.5 and activation function as sigmoid. Initial weights are given as $V_{11}=0.2$, $V_{12}=0.1$, $V_{21}=0.1$, $V_{22}=0.3$, $V_{11}=0.2$, $W_{11}=0.5$, $W_{21}=0.2$



2. Draw the architecture of a multi-layer perceptron.
3. Derive update rules for parameters in the multi-layer neural network through the gradient descent.

Course Outcome 4 (CO4):

1. Give two benefits of using convolutional layers instead of fully connected ones for visual tasks.
2. Suppose that a CNN was trained to classify images into different categories. It performed well on a validation set that was taken from the same source as the training set but not on a testing set. What could be the problem with the training of such a CNN? How will you ascertain the problem? How can those problems be solved?
3. Explain how the cell state is updated in the LSTM model from C_{t-1} to C_t
4. Show the steps involved in an LSTM to predict stock prices.

Course Outcome 5 (CO5):

1. Explain how the cell state is updated in the LSTM model from C_{t-1} to C_t
2. Show the steps involved in an LSTM to predict stock prices.
3. Illustrate the workings of the RNN with an example of a single sequence defined on a vocabulary of four words.

Course Outcome 6 (CO6):

1. Development a deep learning solution for problems in the domain i) natural language processing or ii Computer vision (Assignment)
2. Illustrate the workings of the RNN with an example of a single sequence defined on a vocabulary of four words.

Model Question Paper

QP CODE:

PAGES:4

Reg No: _____

Name: _____

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY
SIXTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR**

Course Code: CST 384

Course Name: CONCEPTS IN DEEP LEARNING

Max. Marks:100

Duration: 3 Hours

PART A

Answer all Questions. Each question carries 3 Marks

1. Distinguish between supervised learning and Reinforcement learning. Illustrate with an example.
2. Differentiate classification and regression.
3. Compare overfitting and underfitting. How it can affect model generalization.

4. Why does a single perceptron cannot simulate simple XOR function? Explain how this limitation is overcome?
5. Illustrate the strengths and weaknesses of convolutional neural networks.
6. Illustrate convolution and pooling operation with an example
7. How many parameters are there in AlexNet? Why the dataset size (1.2 million) is important for the success of AlexNet?
8. Explain your understanding of unfolding a recursive or recurrent computation into a computational graph.
9. Illustrate the use of deep learning concepts in Speech Recognition.
10. What is an autoencoder? Give one application of an autoencoder

(10x3=30
)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.” What is your understanding of the terms task, performance and experience. Explain with two example (10)
- (b) “How does bias and variance trade-off affect machine learning algorithms? (4)

OR

12. (a) Illustrate the concepts of Web search, Page Ranking, Recommender systems with suitable examples. (10)
- (b) List and discuss the different hyper parameters used in fine tuning the (4)

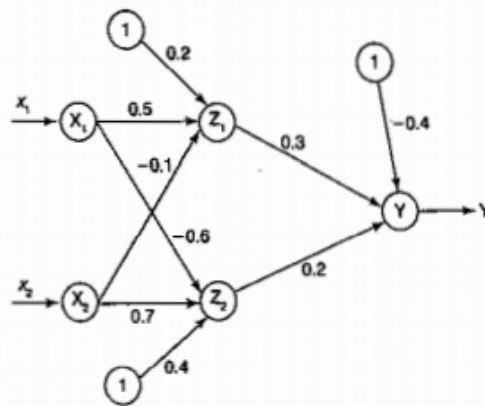
traditional machine learning models

13. (a) How multilayer neural networks learn and encode higher level features from input features. (7)

- (b) Explain gradient decent and delta rule? Why stochastic approximation to gradient descent is needed? (7)

OR

14. (a) Find the new weights for the network using backpropagation algorithm, the network is given with a input pattern $[-1,1]$ and target output as $+1$, Use learning rate of $\alpha=0.3$ and bipolar sigmoid function. (7)



- (b) Write an algorithm for backpropagation which uses stochastic gradient descent method. Comment on the effect of adding momentum to the network. (7)
15. (a) Input to CNN architecture is a color image of size $112 \times 112 \times 3$. The first convolution layer comprises of 64 kernels of size 5×5 applied with a stride of 2 and padding 0. What will be the number of parameters? (5)
- (b) Let $X = [-1, 0, 3, 5]$ $W = [0.3, 0.5, 0.2, 0.1]$ be the the input of i^{th} layer of a neural network and to apply softmax function. What should be the output of it? (4)
- (c) Draw and explain the architecture of convolutional network (5)

OR

16. (a) Explain the concept behind i) Early stopping ii) dropout iii) weight decay (9)

(b) How backpropagation is used to learn higher-order features in a convolutional Network? (5)

17. (a) Explain the working of RNN and discuss how backpropagation through time is used in recurrent networks. (8)

(b) Describe the working of a long short term memory in RNNs. (6)

OR

18. (a) What is the vanishing gradient problem and exploding gradient problem? (8)

(b) Why do RNNs have a tendency to suffer from exploding/vanishing gradient? How to overcome this challenge? (6)

19. (a) Explain any two word embedding techniques (8)

(b) Explain the merits and demerits of using Auto encoders in Computer Vision. (6)

OR

20. (a) Illustrate the use of representation learning in object classification. (7)

(b) Compare Boltzmann Machine with Deep Belief Network. (7)

Teaching Plan

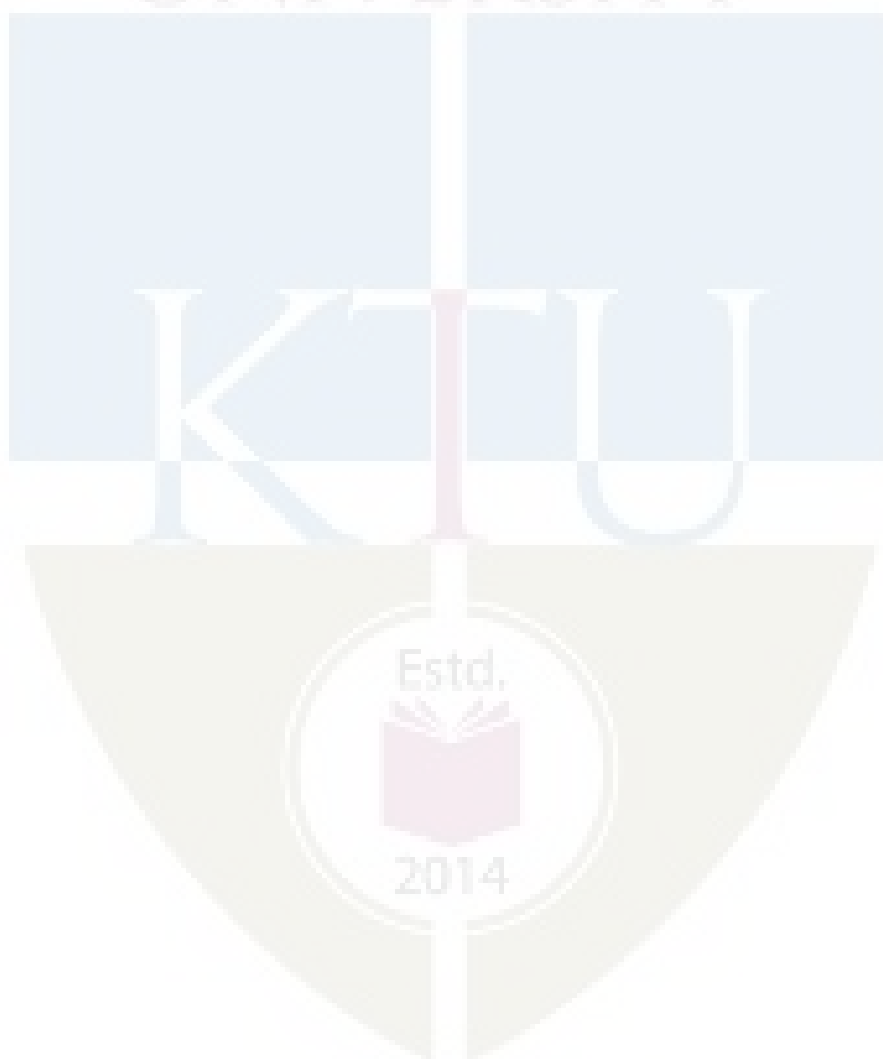
CONCEPTS IN DEEP LEARNING (45 Hours)		
Module 1 : Introduction (9 hours)		
1.1	Key components - Data, models, objective functions, optimization algorithms. (TB2: Section 1.1-1.2)	1 hour

1.2	Learning algorithm (TB1: Section 5.1), Supervised learning- regression, classification (TB2: Section 1.3.1)	1 hour
1.3	tagging, web search, page ranking (TB2: Section 1.3.1)	1 hour
1.4	Recommender systems, Sequence learning, Unsupervised learning, Reinforcement learning(TB2: Section 1.3.2-1.3.4)	1 hour
1.5	Historical Trends in Deep Learning (TB1: Section 1.2).	1 hour
1.6	Concepts: over-fitting, under-fitting, hyperparameters and validation sets. (TB1: Section 5.2-5.3)	1 hour
1.7	Concepts: Estimators, bias and variance. (TB1: Section 5.4)	1 hour
1.8	Demonstrate the concepts of supervised learning algorithms using a suitable platform.	1 hour
1.9	Demonstrate the concepts of unsupervised using a suitable platform.	1 hour
Module 2 : Optimization and Neural Networks (9 hours)		
2.1	Perceptron, Stochastic Gradient descent, Gradient descent solution for perceptron (TB3: Section 1.1 - 1.2.1)	1 hour
2.2	Multilayer perceptron (TB3: Section 1.2.2), (TB1: Section 6.1,6.3)	1 hour
2.3	Activation functions- Sigmoid, tanh, Softmax, ReLU, leaky ReLU (TB3: Section 1.2.1.3 - 1.2.1.5)	1 hour
2.4	Architecture design (TB1: Section 6.4, TB3: Section 1.6)	1 hour
2.5	Chain rule, back propagation (TB3: Section 1.3)	1 hour

2.6	Gradient based learning (TB1: Section 6.2)	1 hour
2.7	Gradient based optimization (TB1: Section 4.3)	1 hour
2.8	Linear least squares using a suitable platform. (TB1: Section 4.5)	1 hour
2.9	Building ML Algorithms and Challenges (TB3: 1.4, TB1: 5.10-5.11)	1 hour
Module 3 :Convolution Neural Network (10 hours)		
3.1	Convolution operation, Motivation, pooling (TB1:Section 9.1-9.3)	1 hour
3.2	Structure of CNN (TB3: Section 8.2)	1 hour
3.3	Convolution and Pooling as an infinitely strong prior (TB1: Section 9.4)	1 hour
3.4	Variants of convolution functions – multilayer convolutional network, tensors, kernel flipping, downsampling, strides and zero padding. (TB1: Section 9.5)	1 hour
3.5	Variants of convolution functions - unshared convolutions, tiled convolution, training different networks. (TB1: Section 9.5)	1 hour
3.6	Structured outputs, data types (TB1: Section 9.6-9.7)	1 hour
3.7	Efficient convolution algorithms. (TB1: Section 9.8,9.10)	1 hour
3.8	Practical challenges of common deep learning architectures- early Stopping (TB3: 4.6)	1 hour
3.9	Practical challenges of common deep learning architectures- parameter sharing, drop-out (TB3: Section 4.9, 4.5.4)	1 hour
3.10	Case Study: AlexNet,VGG, ResNet. (TB3: Section 8.4.1-8.4.3,8.4.5)	1 hour

Module 4 :Recurrent Neural Network (8 hours)		
4.1	Computational graphs (TB1: Section 10.1)	1 hour
4.2	RNN (TB1: Section 10.2-10.3)	1 hour
4.3	Encoder – decoder sequence to sequence architectures. (TB1: Section 10.4)	1 hour
4.4	Deep recurrent networks (TB1: Section 10.5)	1 hour
4.5	Recursive neural networks , Modern RNNs, (TB1: Section 10.6, 10.10)	1 hour
4.6	LSTM and GRU (TB1: Section 10.10, TB3: Section 7.5-7.6)	1 hour
4.7	Practical use cases for RNNs. (TB1: Section 11.1-11.4)	1 hour
4.8	Demonstrate the concepts of RNN using a suitable platform.	1 hour
Module 5 : Applications and Research (9 hours)		
5.1	Computer vision. (TB1: Section 12.2)	1 hour
5.2	Speech recognition. (TB1: Section 12.3)	1 hour
5.3	Natural language processing. (TB1: Section 12.4)	1 hour
5.4	Common Word Embedding -: Continuous Bag-of-Words, Word2Vec (TB3: Section 2.6)	1 hour
5.5	Common Word Embedding -: Global Vectors for Word Representation(GloVe) (TB3: Section 2.9.1- Pennigton 2014)	1 hour
5.6	Brief introduction on current research areas- Autoencoders, Representation learning. (TB3: Section 4.10)	1 hour

5.7	Brief introduction on current research areas- representation learning. (TB3: Section 9.3)	1 hour
5.8	Brief introduction on current research areas- Boltzmann Machines, Deep belief networks. (TB1: Section 20.1, TB3 Section 6.3)	1 hour
5.9	Brief introduction on current research areas- Deep belief networks. (TB1: Section 20.3)	1 hour



CST 385	CLIENT SERVER SYSTEMS	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

The syllabus is prepared with the view of preparing the Engineering Graduates to build effective Client/Server applications. This course aims at providing a foundation in decentralized computer systems, using the client/server model. The course content is decided to cover the essential fundamentals which can be taught within the given slots in the curriculum.

Prerequisite: **Basic knowledge in Computer**

Course Outcomes: After the completion of the course the student will be able to

	Course Outcomes
CO 1	Identify the basics of client/server systems and the driving force behind the development of client/server systems(Cognitive Knowledge Level: Understand)
CO 2	Outline the architecture and classifications of client/server systems(Cognitive Knowledge Level: Understand)
CO 3	Summarize the client/server network services for an application(Cognitive Knowledge Level: Understand)
CO 4	Identify management services and issues in network (Cognitive Knowledge Level: Understand)
CO 5	Outline the Client/Server technology in respect of databases and Client/Server database architecture (Cognitive Knowledge Level: Understand)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓										✓
CO2	✓	✓										✓
CO3	✓	✓										✓
CO4	✓											✓
CO5	✓	✓										✓

Abstract POs defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and team work
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Life long learning

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Percentage)	Test 2 (Percentage)	
Remember	40	40	40
Understand	40	40	40
Apply	20	20	20
Analyse			
Evaluate			
Create			

Mark distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance : 10 marks

Continuous Assessment Test 1 (for theory, for 2 hrs) : 20 marks

Continuous Assessment Test 2 (for lab, internal examination, for 2hrs) : 20 marks

Internal Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 5 questions with 2 questions from each module (2.5 modules x 2 = 5), having 3 marks for each question. Students should answer all questions. Part B also contains 5 questions with 2 questions from each module (2.5 modules

x 2 = 5), of which a student should answer any one. The questions should not have sub-divisions and each one carries 7 marks.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Computing in client server architecture over Mainframe architecture has certain advantages and disadvantages. Describe at least three advantages and disadvantages for each architecture.

Course Outcome 2 (CO2):

1. Explain the role of mainframe-centric model in Client/Server computing?

Course Outcome 3(CO3):

1. Describe the client server system development methodology? Explain different phases of System Integration Life-Cycle.

Course Outcome 4 (CO4):

1. Explain about network management and remote system management. How can security be provided to the network?

Course Outcome 5 (CO5):

1. Explain various types of Client/Server Database Architecture

Syllabus

Module – 1 (Introduction)

Introduction to Client/Server computing - Basic Client/Server Computing Model, Server for Every Client- File Server, Print Server, Application Server, Mail Server, Directory Services Server, Web Server, Database Server, Transaction Servers. Client/Server-Fat or Thin, Stateless

or Stateful, Servers and Mainframes, Client/Server Functions. Driving Forces behind Client/Server Computing- Business Perspective, Technology Perspective.

Module -2 (Client/Server Classification)

Client/Server Types-Single Client/Single Server, Multiple Clients/Single Server, Multiple Clients/Multiple Servers, Integration With Distributed Computing, Alternatives To Client/Server Systems. Classification of Client/Server Systems- Two-Tier Computing, Middleware, Three-Tier Computing- Model View Controller (MVC), Principles behind Client/Server Systems. Client/Server Topologies. Existing Client/Server Architecture. Architecture for Business Information System.

Module -3 (Client/Server Application Components)

Client- Services, Request for services, RPC, Windows services, Print services, Remote boot services, other remote services, Utility Services. Server- Detailed server functionality, Network operating system, Available platforms, Server operating system. Organizational Expectations, Improving performance of client/server applications, Single system image, Downsizing and Rightsizing, Advantages and disadvantages of Client/Server computing, Applications of Client/Server.

Module -4 (Client/ Server Systems Services and Support)

Services and Support- System administration, Availability, Reliability, Scalability, Observability, Agility, Serviceability. Software Distribution, Performance, Network management. Remote Systems Management- RDP, Telnet, SSH, Security. LAN and Network Management issues.

Module -5(Client/Server Technology and Databases)

Client/Server Technology and Databases - Storing Data, Database System Architectures. Client/Server In Respect Of Databases- Client/Server Databases, Client/Server Database Computing, Database Computing Vs. Mainframe, PC/File Server Computing. Client/Server Database Architecture - Process-Per-Client Architecture, Multi-Threaded Architecture, Hybrid Architecture. Database Middleware Component - Application Programming Interface, Database Translator, Network Translator.

Text Book

1. Patrick Smith & Steve Guengerich, Client / Server Computing, PHI
2. Subhash Chandra Yadav, Sanjay Kumar Singh, An Introduction to Client/Server Computing, New Age International Publishers

Reference Books

1. Jeffrey D.Schank, “Novell’s Guide to Client-Server Application & Architecture” Novell Press
2. Robert Orfali, Dan Harkey, Jeri Edwards, Client/Server Survival Guide, Wiley-India Edition, Third Edition
3. Dawna Travis Dewire, Client Server Computing — McGraw Hill
4. W.H.Inman, Developing Client Server Applications, BPB

Model Question Paper

QP CODE: _____

PAGES: ____

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

FIFTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 385

Course Name : Client Server Systems

Max Marks: 100

Duration: 3 Hours

PART-A

(Answer All Questions. Each question carries 3 marks)

1. Differentiate between Stateful and Stateless servers
2. List the different phases and activities of client/server system development methodology.
3. How does transmission protocol work in client/server applications?
4. List any six services in single system image environment.
5. Specify the role of the client in Client/Server computing and also list any six services provided by the client.
6. Why do most RPC system support call by value semantics for parameter passing?
7. What do you mean by a thin client network? List three advantages of the Thin

Client Network system.

8. How are connectivity and interoperability between .client/server achieved?
9. One disadvantage of the Client/Server system is lack of control in a Database Management environment. Justify.
10. Explain the DBMS concept in client/server architecture.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Differentiate between Transaction server and Data server system with examples. (7)
- (b) Computing in client server architecture over Mainframe architecture has certain advantages and disadvantages. Describe at least three advantages and disadvantages for each architecture. (7)

OR

12. (a) Explain various Clients/Server system development tools. (6)
- (b) Classify and describe the driving forces that drive the move to Client/Server computing. (8)
13. (a) Explain the role of mainframe-centric model in Client/Server computing? (5)
- (b) Describe the three types of Client/Server systems in existence (9)

OR

14. (a) List and explain the general forces behind the architecture for business information systems (7)
- (b) Explain the different distribution styles. (7)
15. (a) Illustrate the concept of rightsizing and downsizing in Client/Server Computing (7)
- (b) What is client server system development methodology? Explain the (7)

different phases of System Integration Life-Cycle.

OR

16. (a) In Client/Server computing, explain the following with examples (10)
- i. Dynamic Data Exchange
 - ii. RPC, Remote Procedure Call
 - iii. Remote Boot Service
 - iv. Diskless Computer
 - v. Object-linking and embedding
- (b) Explain the functions and features of Network Operating System (4)
17. (a) Explain about network management and remote system management. How can security be provided to the network ? (10)
- (b) In client server architecture, what do you mean by Availability, Reliability, Serviceability and Security? Explain with examples. (4)

OR

18. (a) Client server is modular infrastructure, this is intended to improve Usability, Flexibility, Interoperability and Scalability. Explain each term with an example, in each case how it helps to improve the functionality of client server architecture. (7)
- (b) Explain about network management and remote system management. How can security be provided to network? (7)
19. (a) Explain the different types of Client/Server Database Architecture (9)
- (b) List and explain the main components of Database middleware (5)

OR

20. (a) Discuss types of database utilities, tools and their functions (7)
- (b) Discuss about the role of traditional and web databases in handling client/server based applications. (7)

Teaching Plan

Module- 1(Introduction)		(10 hours)
1.1	Basic Client/Server Computing Model	1 hour
1.2	Server for Every Client- File Server, Print Server	1 hour
1.3	Application Server, Mail Server, Directory Services Server	1 hour
1.4	Web Server, Database Server	1 hour
1.5	Transaction Servers	1 hour
1.6	Client/Server-Fat or Thin	1 hour
1.7	Stateless or Stateful	1 hour
1.8	Servers and Mainframes	1 hour
1.9	Client/Server Functions	1 hour
1.10	Driving Forces behind Client/Server Computing- Business Perspective, Technology Perspective	1 hour
Module- 2 (Client/Server Classification)		(10 hours)
2.1	Client/Server Types-Single Client/Single Server	1 hour
2.2	Multiple Clients/Single Server, Multiple Clients/Multiple Servers	1 hour
2.3	Integration With Distributed Computing	1 hour
2.4	Alternatives To Client/Server Systems	1 hour
2.5	Classification of Client/Server Systems- Two-Tier Computing, Middleware	1 hour
2.6	Three-Tier Computing- Model View Controller (MVC)	1 hour
2.7	Principles behind Client/Server Systems.	1 hour
2.8	Client/Server Topologies	1 hour
2.9	Existing Client/Server Architecture	1 hour
2.10	Architecture for Business Information System	1 hour
Module -3 (Client/Server Application Components)		(9 hours)
3.1	The client: Services, Request for services, RPC	1 hour
3.2	Windows services, Print services, Remote boot services	1 hour

3.3	Utility Services & Other Services	1 hour
3.4	Server- Detailed server functionality, Network operating system	1 hour
3.5	Available platforms, Server operating system	1 hour
3.6	Organizational Expectations, Improving performance of client/server applications	1 hour
3.7	Single system image, Downsizing and Rightsizing	1 hour
3.8	Advantages and disadvantages of Client/Server computing	1 hour
3.9	Applications of Client/Server	1 hour
Module -4 (Client/ Server Systems Services and Support)		(8 hours)
4.1	Services and Support, System administration	1 hour
4.2	Availability, Reliability	1 hour
4.3	Scalability, Observability, Agility	1 hour
4.4	Serviceability, Software Distribution	1 hour
4.5	Performance	1 hour
4.6	Network management	1 hour
4.7	Remote Systems Management- RDP, Telnet, SSH	1 hour
4.8	Security, LAN and Network Management issues	1 hour
Module -5(Client/Server Technology and Databases)		(8 hours)
5.1	Client/Server Technology and Databases - Storing Data	1 hour
5.2	Database System Architectures	1 hour
5.3	Client/Server In Respect Of Databases- Client/Server Databases	1 hour
5.4	Client/Server Database Computing	1 hour
5.5	Database Computing Vs. Mainframe, PC/File Server Computing	1 hour
5.	Client/Server Database Architecture - Process-Per-Client Architecture	1 hour
5.7	Multi-Threaded Architecture, Hybrid Architecture	1 hour
5.8	Database Middleware Component - Application Programming Interface, Database Translator, Network Translator	1 hour

CST 386	WIRELESS NETWORKS AND IoT APPLICATIONS	Category	L	T	P	Credit	Year of Introduction
		VAC	3	1	0	4	2019

Preamble:

This course equips the learners with fundamental wireless technologies for the Internet of Things(IoT) and the IoT ecosystem. It covers the underlying concepts in wireless networks, communication mechanisms, protocols, hardware, software, and the cloud platforms for IoT. The students will be able to design smart IoT applications for real world problems..

Prerequisite: Sound knowledge in Data Communication, Computer Networks and Programming in C

Course Outcomes: After the completion of the course the students will be able to

CO1	Recognize wireless technologies required for IoT ecosystem (Cognitive Knowledge Level : Understand)
CO2	Perceive the concept of IoT and M2M architecture, IoT examples, and Data Management in IoT (Cognitive Knowledge Level :Apply)
CO3	Outline the hardware components used in IoT including Sensors, Actuators and development boards (Cognitive Knowledge Level : understand)
CO4	Explain the software components of IoT (Cognitive Knowledge Level :Understand)
CO5	Demonstrate the protocols used in IoT and build IoT Programs (Cognitive Knowledge Level : Apply)
CO6	Build IoT-based smart real-time applications such as Smart Healthcare, Smart Agriculture, Smart Environment and Smart Home (Cognitive Knowledge Level : Apply)

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1												
CO2												

CO3												
CO4												
CO5												
CO6												

Abstract POs Defined by National Board of Accreditation			
PO#	Broad PO	PO#	Broad PO
PO1	Engineering Knowledge	PO7	Environment and Sustainability
PO2	Problem Analysis	PO8	Ethics
PO3	Design/Development of solutions	PO9	Individual and teamwork
PO4	Conduct investigations of complex problems	PO10	Communication
PO5	Modern tool usage	PO11	Project Management and Finance
PO6	The Engineer and Society	PO12	Lifelong learning

Assessment Pattern

Blooms Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Percentage)	Test 2 (Percentage)	
Remember	30	30	30
Understand	50	40	40
Apply	20	30	30

Analyze			
Evaluate			
Create			

Mark Distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 Hours

Continuous Internal Evaluation Pattern:

Attendance	10 marks
Continuous Assessment Tests	25 marks
Continuous Assessment Assignment	15 marks

Internal Examination Pattern:

Each of the two internal examinations has to be conducted out of 50 marks. First Internal Examination shall be preferably conducted after completing the first half of the syllabus, and the Second Internal Examination shall be preferably conducted after completing the remaining part of the syllabus. There will be two parts: Part A and Part B. Part A contains 5 questions (preferably, 2 questions each from the completed modules and 1 question from the partly covered module), having 3 marks for each question adding up to 15 marks for part A. Students should answer all questions from Part A. Part B contains 7 questions (preferably, 3 questions each from the completed modules and 1 question from the partly covered module), each with 7 marks. Out of the 7 questions in Part B, a student should answer any 5.

End Semester Examination Pattern:

There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer anyone. Each question can have a maximum 2 subdivisions and carries 14 marks.

Syllabus

Module- 1 (Introduction to IoT and wireless technologies required for IoT)

Internet of Things, Role of Things and the Internet, Wireless IoT. Wireless Networks - Network Topologies, Types of Networks. Role of Wireless Standards in IoT. Protocol Stack - OSI Model, TCP/IP Model, IEEE 802 Reference Model, Protocols for Wireless IoT. Bluetooth - Transceiver, Frequency Channels, Typical Range, Access and Spread Spectrum, Modulation and Data Rate, Error Correction and Detection, Network Topology. ITU G.9959, Zwave, IEEE 802.15.4, Zigbee Specification, Thread, WiFi, 6LowPAN, IPv6, LoRaWAN.

Module- 2 (IoT architecture, Data and Device management)

Internet of Things - IoT Architectural View, Technology Behind IoT - Server End Technology, Sources of Internet of Things, M2M Communication. IoT Application Areas. IoT Examples. IoT Data Management - Device Management Gateways. Design Principles for Web Connectivity - Web Communication Protocols for Connected Devices, Web Connectivity for Connected Devices using Gateways. Internet Connectivity Principles – Internet Connectivity, Internet based communication, IP addressing in the IoT.

Module- 3 (Data Acquiring and Enabling Technologies)

Data Acquiring and Storage for IoT Services- Organization of Data, Big data, Acquiring Methods, Management Techniques, Analytics, Storage Technologies. Cloud Computing for Data storage - IoT Cloud based Services using Xively, Nimbits, and Other Platforms. Sensor Technologies for IoT Devices - Sensor Technology, Participatory Sensing, Industrial IoT and Automotive IoT, Actuators for Various Devices, Sensor Data Communication Protocols, Wireless Sensor network Technology

Module-4 (Prototyping the Embedded Devices for IoT)

Embedded Computing Basics, Embedded Hardware Unit. Embedded Platforms for Prototyping - Arduino, Intel Galileo, Intel Edison, Raspberry Pi, BeagleBone, mBed. Prototyping and Designing the Software for IoT Applications- Introduction, Prototyping Embedded Device Software- Programming using Arduino, Programming for an Arduino Controlled Traffic Control Lights at a Road Junction, Basic Arduino Programs to Blink LED, Find the Distance using Ultrasonic Sensor, Estimate Room Temperature, Measuring Soil Moisture Level

Module 5 (Business Models and Case Studies)

Business Models and Processes using IoT. Value Creation in the Internet of Things. Cloud PaaS- Xively, Nimbits, IBM Bluemix, CISCO IoT, AWS IoT, TCS Connected AWS Platform, Case studies- Smart Home, Smart Environment, Smart healthcare, Smart agriculture

Text Books

1. Daniel Chew, “Wireless Internet of Things -A Guide to the lower layers”, IEEE Standards and Association, IEEE Press, Wiley
2. Rajkamal, “Internet of Things : Architecture and Design Principles”, McGraw Hill (India) Private Limited.

References

1. ArshadeepBahga, Vijay Madiseti, “Internet of Things: A hands-on approach”, University Press, 2015 (First edition)
2. Dieter Uckelmann, Mark Harrison, Michahelles Florian (Ed.), Architecting the internet of things, Springer, 2011
3. Dr. Ovidiu Vermesan, Dr. Peter Friess, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers, 2013
4. Simon Monk, “Programming Arduino: Getting Started with Sketches”, McGraw Hill Publications

Sample Course Level Assessment Questions**Course Outcome 1 (CO1):**

1. Compare Bluetooth and Bluetooth LE power classes
2. Demonstrate Zigbee Specification Protocol Stack

Course Outcome 2 (CO2):

1. What are the major components of IOT system? Briefly explain each
2. Correlate M2M architectural Levels with IOT architectural Levels

Course Outcome 3 (CO3):

1. Describe the use of GPIO pins ?
2. What are actuators ? Mention the roles of actuators in IoT systems

Course Outcome 4(CO4):

1. Identify the role of HBase in Hadoop File System
2. Differentiate Edge computing and Distributed computing
3. Illustrate open protocols, tools and frameworks generally used in M2M

Course Outcome 5(CO5):

1. What do you mean by Arduino sketches?
2. Write an Arduino program to blink LED

Course Outcome 6(CO6):

1. How IoT technology helps TELEMEDICINE in India?
2. How soil moisture can be detected in Smart Agriculture?

Model Question Paper

QP CODE:

PAGES :2

Reg No: _____

Name: _____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE EXAMINATION(MINOR), MONTH & YEAR

Course Code: CST 386

Course Name: WIRELESS NETWORKS AND IoT APPLICATIONS

Max.Marks:100

Duration: 3 Hours

PART A

Answer All Questions. Each Question Carries 3 Marks

1. Illustrate Role of *things* and *internet* in IoT
2. What is Bluetooth? Explain the range and frequency channels of Bluetooth?
3. List any three the features of Constrained Application Protocol (COAP).
4. Compare Raspberry Pi and BeagleBoard boards.
5. Identify the role of HBase in Hadoop File System.
6. Differentiate Edge computing and Distributed computing.
7. Give an example of Raspberry Pi applications for Industrial IoT.
8. What are the on-board functional units in Intel Galileo?
9. Interpret the concept of value creation in IoT.

10. Explain the use of PaaS in IoT Smart applications with any three examples.

(10x3=30)

Part B

(Answer any one question from each module. Each question carries 14 Marks)

11. (a) Compare various Network topologies used in Wireless Networks. (8)
- (b) Describe the following wireless technologies on i) *Zigbee* ii) *WiFi* iii) *Thread*. (6)

OR

12. (a) Explain protocol stacks used in wireless networks for IoT applications. (8)
- (b) Illustrate the Architectural design of LoRaWAN. (6)
13. (a) Define M2M. Explain M2M architecture. Correlate M2M architectural levels with IoT architectural levels. (8)
- (b) Compare SOAP and REST protocols. (6)

OR

14. (a) Summarize different Online Transactions and Processing techniques. (8)
- (b) Identify the functions of Device-Management Gateway. (6)
15. (a) Define actuators ? Describe the roles of actuators in IoT systems. (8)
- (b) Explain the usage contexts of analog sensors and digital sensors. (6)

OR

16. (a) How data collection, storage & computing services done using Nimbits? (10)
- (b) List any four features of Xively. (4)

17. (a) What do you mean by Arduino sketches? (4)
- (b) Write an Arduino program to blink LED (10)

OR

18. (a) Demonstrate an example of Raspberry Pi applications for Industrial IoT. (10)
- (b) Compare the features of Arduino-R3 and Arduino Yun boards. (4)
19. (a) Explain various tasks of a smart irrigation monitoring service. (8)
- (b) Demonstrate the tasks of Soil-Moisture monitoring service. (6)

OR

20. (a) a) Mr. Kiran Mathew has been a chronic diabetic patient for the past few years. He was under regular check up at the hospital every two weeks. All of a sudden the pandemic like COVID-19 arises in the country and the government issues a lockdown for a period of two months. Illustrate how Mr. Kiran can be monitored by the health care worker using intelligent healthcare techniques. (10)
- (b) Mention any four sensors used in smart healthcare (4)

TEACHING PLAN

No	Contents	No of Lecture Hrs(45)
Module – 1 (Introduction to IoT and wireless technologies required for IoT) (8 hrs) (TB-1, Chapter 1...)		
1.1	Internet Of Things, Role of things and internet ,Wireless IoT	1
1.2	Wireless Networks- Network Topologies-Types of Networks,Role of	1

	Wireless standards in IoT	
1.3	Protocol Stack-OSI Model- TCP/IP Model-IEEE 802 reference model	1
1.4	Protocols for Wireless IoT-Bluetooth-Transceiver, Frequency Channels-Typical Range, Access and Spread Spectrum, Modulation and Data Rate	1
1.5	Error Correction and Detection-Network Topology.	1
1.6	ITU G.9959, Zwave, IEEE 802.15.4, Zigbee Specification	1
1.7	Thread, Wifi, 6LowPAN, IPv6	1
1.8	LoRaWAN	1
Module- 2 (IOT architecture, Data and Device management) (9hrs)		
2.1	Internet of Things -IoT Architectural view	1
2.2	Technology Behind IOT-Server End Technology,Sources of Internet of Things	1
2.3	M2M Communication.	1
2.4	IoT Application Areas. IOT Examples.	1
2.5	IoT Data Management, Device Management Gateways.	1
2.6	Design Principles for Web Connectivity	1
2.7	Web communication protocols for connected devices,	1
2.8	Web connectivity for connected devices using Gateways.	1
2.9	Internet connectivity Principles – Internet Connectivity, Internet based communication, IP addressing in the IoT.	1
Module- 3 (Data Acquiring and Enabling Technologies (8 hrs)		
3.1	Data acquiring and storage for IoT devices- Organization of Data, Big data	1
3.2	Acquiring methods, management techniques, Analytics, Storage technologies.	1
3.3	Cloud computing for Data storage-IoT Cloud based services using Xively,	1

	Nimbits, and other platforms.	
3.4	Cloud computing-Nimbits	1
3.5	Sensor Technologies for IoT Devices-Sensor Technology, Participatory sensing	1
3.6	Industrial IoT and Automotive IoT	1
3.7	Actuators for various devices, Sensor data communication protocols	1
3.8	Wireless Sensor network Technology	1
Module 4(Prototyping the Embedded Devices for IoT)(9hrs)		
4.1	Introduction, Embedded Computing Basics, Embedded Hardware Unit.	1
4.2	Embedded Platforms for Prototyping-Arduino, Intel Galileo	1
4.3	Intel Edison, Raspberry Pi, BeagleBone, mBed	1
4.4	Prototyping and designing the software for IoT applications-Introduction, Prototyping embedded device software	1
4.5	Prototyping and designing the software for IoT applications-Introduction, Prototyping embedded device software	1
4.6	Programming concepts in Arduino	1
4.7	Programming for an arduino controlled traffic control lights at a road junction	1
4.8	Basic Arduino programs to blink LED, Find the distance using ultrasonic sensor	1
4.9	Estimate room temperature, Measuring soil moisture level	1
Module 5 (higher level protocols and case studies)(9 hrs)		
5.1	Business Models and Processes using IOT, Value creation in the Internet of Things.	1

5.2	Xively, Nimbits, IBM Bluemix	1
5.3	CISCO IoT, AWS IoT, TCS Connected AWS Platform	1
5.4	Case Study- Smart Environment	1
5.5	Case Study- Smart Environment	1
5.6	Case study Smart Home	1
5.7	Case study Smart Home	1
5.8	Case study Smart healthcare (Lecture I)	1
5.9	Case study Smart healthcare (Lecture II)	1
5.10	Case study -Smart agriculture (Lecture I)	1
5.11	Case study -Smart agriculture (Lecture II)	1

