Course: BTech **Semester:** 3

Prerequisite: Basic knowledge of software applications |

Rationale: This course provides a broad introduction to software engineering. The various process models required to develop software is also being described. Moreover the functional and non-functional requirements are also described.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Design introduction:** Object-oriented programming, oops principles, encapsulation, inheritance and polymorphism java as a oops & internet enabled language, importance of java, java usage in industry, the byte code, compiling, and running of simple java program, jvm, jdk, jre	8	4
2	**Data types, variable, operators:** Data types, variables, dynamic initialization, scope and lifetime of variables, type conversion and casting, operators	10	4
3	**Control statements:**Conditional Statements, Looping Statements, Jump Statements	10	5
4	**Arrays:**Array, Array values and memory storage Structure, Types of Arrays.	8	4
5	**Object oriented programming:** Classes and objects: concepts of classes and objects, declaring objects, assigning object reference variables, methods, constructors, access control, garbage collection, usage of static with data and methods, usage of final with data, overloading methods and constructors, parameter passing - call by value, recursion, nested classes.	18	9
6	**Inheritance:** Inheritance Basics, member access rules, Usage of super key word, forms of inheritance, Method Overriding, Abstract classes, Dynamic method dispatch, Using final with inheritance	8	2
7	**Strings, Packages and Interfaces:** String handling functions, Packages, Class path, importing packages, differences between classes and interfaces, Implementing & Applying interface, enumerations in java.	12	5
8	**Exception Handling:**Exceptions, Types of Exceptions, Handling of Exceptions	8	3
9	**Multi Threading:**Thread, Usage of threads, Types of threads, Handling Threads	10	4
10	**Collections Framework:**Functional Programming, Collections, Hierarchy of collections	5	8

Reference Books

1.	**Introduction to Java Programming (Comprehensive Version) Daniel Liang; Pearson**
2.	**Core Java Volume-II Fundamentals Horstmann & Cornell; Pearson**
3.	**Complete Reference Java 2 Herbert Schildt; TMH**

Course Outcomes

At the end of this course Students Will be able to:	
1	Describe the procedural and object oriented paradigm with concepts of streams, classes, functions, data and objects
2	Understand dynamic memory management techniques using pointers, constructors, destructors, etc
3	Describe the concept of function overloading, operator overloading, virtual functions and polymorphism
4	Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming.
5	Demonstrate the use of various OOPs concepts with the help of programs

Course Outcome

After Learning the Course the students shall be able to:
After learning the course the students shall be able to: 1. Understand the principles and practice of object oriented programming. 2. Write, compile and debug programs with Java compiler. 3. Create a robust application using exception handling. 4. Understand the principles of synchronization and design application using multi-threading.

Course: BTech **Semester:** 3

Prerequisite: Basic knowledge of software applications |

Rationale: This course provides a broad introduction to software engineering. The various process models required to develop software is also being described. Moreover the functional and non-functional requirements are also described.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	Internal Marks			External Marks		
					T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Outcome

After Learning the Course the students shall be able to:
After learning the course the students shall be able to: 1. Understand the principles and practice of object oriented programming. 2. Write, compile and debug programs with Java compiler. 3. Create a robust application using exception handling. 4. Understand the principles of synchronization and design application using multi-threading.

List of Practical

1.	**write a program to display Hello World message in console window.**
2.	**Write a program to perform arithmetic and bitwise operations in a single source program without object creation.**
3.	**Write a program to perform arithmetic and bitwise operations by creating individual methods and classes than create an object to execute the individual methods of each operation.**
4.	**Write a java program to display the employee details using Scanner class.**
5.	**Write a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions?**
6.	**The Fibonacci sequence is defined by the following rule. The first 2 values in the sequence are 1, 1. Every subsequent value is the sum of the 2 values preceding it. Write a Java program that uses both recursive and non- recursive functions to print the nth value of the Fibonacci sequence?**
7.	**Write a Java program that prompts the user for an integer and then prints out all the prime numbers up to that Integer?**
8.	**Write a Java program to multiply two given matrices?**
9.	**Write a Java program for sorting a given list of names in ascending order?**

10.	Write a java program for Method overloading and Constructor overloading
11.	Write a java program to represent Abstract class with example.
12.	Write a program to implement multiple Inheritances.
13.	write program to demonstrate method overriding and super keyword.
14.	Write a java program to implement Interface using extends keyword.
15.	Write a java program to create inner classes.
16.	Write a java program to create user defined package.
17.	Write a Java program that displays the number of characters, lines and words in a text?
18.	Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome?
19.	Write a Java program that reads a line of integers and then displays each integer and the sum of all integers. (Use StringTokenizer class)?
20.	Write a java program for creating single try block with multiple catch blocks.
21.	write a program for multiple try blocks and multiple catch blocks including finally.
22.	write a program to create user defined exception.
23.	Write a java program for producer and consumer problem using Threads.
24.	Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
25.	write a program to create dynamic array using ArrayList class and the print the contents of the array object.
26.	Write programs to implement add, search and remove operation on ArrayList object.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc.