Course: BTech **Semester:** 3

Prerequisite: Basic knowledge of software applications

Course Objective: 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	
2	0	0	0	2	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

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 (TextBook)**
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.