S. No	Course Code	Name of the Course	L	T	P	C
5	19CST302/ 19ITT302	Java Programming (Common to CSE & IT Branches)	2	1	0	3

Pre-Requisites: Knowledge on Computer Hardware, Basic knowledge of problem solving.

Course objectives:

- Problem solving using object-oriented paradigm.
- Capacity of creating a good release strategy.
- Capable of writing thread safe solutions for better throughput.
- GUI Application building by handling events.
- Capacity to write applications that team up users to communicate.

Unit No	Contents	Mapped CO
I	**Introduction to OOPS Concepts, Classes and Strings 10 hrs** Introduction to Object Oriented Programming, Java buzzwords, Java Programming Basics, Sample programs, Data types and operators, Control statements. Classes: Classes, Objects, Methods, Constructors, this and static keywords, Method and Constructor Overloading, Access modifiers, arrays-One Dimensional and multi dimensional arrays, Searching, Sorting. Strings-Exploring the String class, String buffer class, Command- line arguments.	CO1
II	**Inheritance, Interfaces, Packages And Exceptions 09 hrs** Inheritance: Need of inheritance, types, and super keyword, abstract classes, interfaces, compile time and runtime polymorphism, Packages. Exception Handling: Concepts of Exception handling, Built-in exceptions, creating own exception sub classes, Assertions.	CO2
III	**Multi-Threading and I/O Streams 10 hrs** Multithreading : Concepts of Multithreading, differences between process and thread, thread life cycle, Thread class, Runnable interface, creating multiple threads, Synchronization, thread priorities, inter thread communication, daemon threads, thread groups. Stream based I/O (java.io) – The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, Random access file operations, Object Serialization, exploring java.io.	CO-3

IV	**Collection Frame Work Classes** **10 hrs** The Collections Framework (java.util)- Collections overview, Collection Interfaces, The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Accessing a Collection via an Iterator, Using an Iterator, The For-Each alternative, Map Interfaces and Classes, Comparators, Collection algorithms, Arrays, The Legacy Classes and Interfaces- Dictionary, Hashtable, Properties, Stack, Vector.	**CO4**
V	**GUI Programming and Networking** **09 hrs** GUI Programming with Swing: Introduction, limitations of AWT, Various swing components & hierarchy. Event Handling- event delegation model, sources of event, Event Listeners, adapter classes, inner classes. Introduction to Networking: Basics of Networking, Networking classes and Interfaces, Networking with URLs, exploring java.net package.	**CO5**
Advanced topics in this Course: 1. Unit-3: Collection API 2. Unit-4: Network Programming using java.net 3. As part of lab, JUnit for testing applications.		

Course Outcomes:

CO-1: Implement object oriented programming concepts, and apply them in solving problems. **(Apply)**

CO-2: Experiment the implementation of packages and interfaces. (Apply)

CO-3: Experiment the concept of multithreading over single threaded programming. **(Analyze)**

CO-4: Use generic data structures of collection framework to manage data. **(Apply)**

CO-5: Test the GUI based network applications among multiple users through network programming. **(Analyze**)

Text books:

1. Java- The Complete Reference, Herbert Schildt, MC GRAW HILL Education, 9th Edition, 2016.

Reference books:

1. "Java – How to Program", Paul Deitel, Harvey Deitel, PHI, 9th Edition, 2012
2. "Core Java", Nageswara Rao, Wiley Publishers.
3. Thinking in Java", Bruce Eckel, Pearson Education., 4th Edition
4. "A Programmers Guide to Java SCJP", 3rd Edition, Mughal, Rasmussen, Pearson Education

e- Resources & other digital material:

1. Programming in Java: https://nptel.ac.in/courses/106/105/106105191/
2. Oracle Java API 7 : https://docs.oracle.com/javase/7/docs/api/

CO-PO mapping Table with Justification

Contribution of Course Outcomes towards achievement of Program Outcomes &strength of correlations (High: 3, Medium: 2, Low:1)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO12	PSO1	PSO2
CO1	2	-	2	-	3	1	-	2	2	2	-	-	2	2
CO2	2	-	2	-	3	1	-	2	2	2	-	-	2	2
CO3	2	2	2	-	3	1	-	2	2	2	-	-	2	2
CO4	2	-	2	-	3	1	-	2	3	2	-	2	2	2
CO5	2	2	2	-	3	1	-	2	3	2	-	2	2	2

Justification:

Mapping	Score	Justification
CO1-PO1, CO2-PO1 CO3-PO1, CO4-PO1 CO5-PO1	2	Problems based on science and engineering fundamentals are worked out.
CO3-PO2, CO5-PO2	2	Problem solving based on system architecture and network based applications are practiced.
CO1-PO3, CO2-PO3 CO3-PO3, CO4-PO3 CO5-PO3	2	Develop solutions for real time problems.
CO1-PO5, CO2-PO5 CO3-PO5, CO4-PO5 CO5-PO5	3	Solving problem using IDE's like Eclipse/Netbeans with plugins like JUnit, Code Coverage etc.
CO1-PO6, CO2-PO6 CO3-PO6, CO4-PO6 CO5-PO6	2	Problem solving following the clean code practices.
CO1-PO8, CO2-PO8 CO3-PO8, CO4-PO8 CO5-PO8	2	Problem solving following Ethical Practices.

CO1-PO9, CO2-PO9 CO3-PO9	2	Collaborative working in problem solving.
CO4-PO9, CO5-PO9	3	Collaborative working in problem solving.
CO1-PO10, CO2-PO10 CO3-PO10, CO4-PO10 CO5-PO10	2	Interact with faculty and peers for investigating the problems to be solved and document the same.
CO4-PO9 CO5-PO9	2	Explore vendor specific API for latest libraries.
CO1-PSO1, CO2-PSO1 CO3-PSO1, CO4-PSO1 CO5-PSO1	2	Problem solving with ease through best software engineering.
CO1-PSO2, CO2-PSO2, CO3-PSO2 CO4-PSO2, CO5-PSO2	2	Language learned should become a skill to be successful in planning the career.

*****