

GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)**Competency-focused Outcome-based Green Curriculum-2021 (COGC-2021)**

Semester-IV

Course Title: Java Programming

(Course Code: 4343203)

Diploma programme in which this course is offered	Semester in which offered
Information and Communication Technology	Fourth

1. RATIONALE

This course is designed to teach object-oriented programming concepts, techniques, and applications using the Java programming language. Object-oriented programming emphasis on the fundamentals of the structured design with classes, including development, testing, implementation and documentation also includes object-oriented programming techniques, classes and objects. Java is a simple, portable, distributive, robust, secure, dynamic, architecture neutral, object oriented programming language. Java programming language is designed to enable the development of a small, reliable, portable, distributed, real-time operating platform, high-performance applications for the widest range of computing platforms possible as well as cross-platform interaction. By making applications available across heterogeneous environments, businesses can provide more services, boost end-user productivity, communication and collaboration to enterprise and consumer applications. The Java programming language originated as part of a research project to develop advanced software for a wide variety of network devices and embedded systems. The Java programming language is used as the teaching vehicle for this course.

The aim of this course is that student should learn platform independent object oriented programming and java as base language for advanced technology like three tier architecture applications, cloud computing and web development. Java is also used to program electronic hardware devices such as air conditioners, televisions, refrigerators, washing machines and many others.

2. COMPETENCY

The aim of this course is to help the students to attain the following industry identified competency through various teaching-learning experiences:

- Develop java application using object-oriented approach.

3. COURSE OUTCOMES (COs)

The practical exercises, the underpinning knowledge and the relevant soft skills associated with this competency are to be developed in the student to display the following COs:

The practical experiences and relevant soft skills associated with this course are to be taught and implemented, so that the student demonstrates the following industry-oriented COs associated with the above-mentioned competency:

Course Outcomes:

- Write simple java programs for a given problem statement.
- Use object oriented programming concepts to solve real world problems.
- Develop an object-oriented program using inheritance and package concepts for a given problem statement.
- Develop an object oriented program using multithreading and exception handling for a given problem statement.
- Develop an object-oriented program by using the files and collection framework.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme (In Hours)			Total Credits (CI+T/2+P/2)	Examination Scheme				
				Theory Marks		Practical Marks		Total Marks
CI	T	P	C	CA	ESE	CA	ESE	
3	0	4	5	30	70	25	25	150

Out of 30 marks under the theory CA, 10 marks are for assessment of the micro-project to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessing the attainment of the cognitive domain UOs required for the attainment of the COs.

Legends: CI-Class Room Instructions; T – Tutorial/Teacher Guided Theory Practice; P - Practical; C – Credit, CA - Continuous Assessment; ESE - End Semester Examination.

5. SUGGESTED PRACTICAL EXERCISES

The following practical outcomes (PrOs) that are the sub-components of the COs. Some of the **PrOs** marked '*' are compulsory, as they are crucial for that particular CO. These PrOs need to be attained at least at the 'Precision Level' of Dave's Taxonomy related to 'Psychomotor Domain'.

Sr. No	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Req
1.	* Install JDK, write a simple "Hello World" or similar java program, compilation, debugging, executing using java compiler and interpreter.	I	2
2.	Write a program in Java to find maximum of three numbers using conditional operator.	I	1
3.	Write a program in Java to reverse the digits of a number using while loop	I	1

4.	Write a program in Java to add two 3*3 matrices.	I	2
5.	Write a program in Java to generate first n prime numbers.	I	2
6.	Write a program in Java which has a class Student having two instance variables enrollmentNo and name. Create 3 objects of Student class in main method and display student's name.	II	1
7.	*Write a program in Java which has a class Rectangle having two instance variables height and weight. Initialize the class using constructor.	II	1
8.	Write a program in Java demonstrate the use of "this" keyword.	II	2
9.	Write a program in Java to demonstrate the use of "static" keyword.	II	2
10.	Write a program in Java to demonstrate the use of "final" keyword.	II	2
11.	*Write a program in Java which has a class Shape having 2 overloaded methods area(float radius) and area(float length, float width). Display the area of circle and rectangle using overloaded methods.	II	2
12.	Write a program in Java to demonstrate the constructor overloading.	II	2
13.	Write a java program to demonstrate use of "String" class methods : chatAt(), contains(), format(), length(), split()	II	1
14.	Write a program in Java to demonstrate single inheritance	III	1
15.	Write a program in Java to demonstrate multilevel inheritance	III	2
16.	Write a program in Java to demonstrate hierarchical inheritance.	III	2
17.	Write a program in Java to demonstrate method overriding.	III	2
18.	*Write a program in Java which has a class Car having two instance variables topSpeed and name. Override toString() method in Car class. Create 5 instances of Car class and print the instances.	III	2
19.	Write a program in Java to implement multiple inheritance using interfaces.	III	2
20.	*Write a program in Java which has an abstract class Shape having three subclasses: Triangle , Rectangle , and Circle . Define method	III	4

	area() in the abstract class Shape and override area() method to calculate the area.		
21.	Write a program in Java to demonstrate use of final class.	III	2
22.	Write a program in Java to demonstrate use of package.	III	2
23.	Write a program in Java to develop user defined exception for 'Divide by Zero' error.	IV	2
24.	*Write a program in Java to develop Banking Application in which user deposits the amount Rs 25000 and then start withdrawing of Rs 20000, Rs 4000 and it throws exception "Not Sufficient Fund" when user withdraws Rs. 2000 thereafter.	IV	2
25.	*Write a program that executes two threads. One thread displays "Thread1" every 1000 milliseconds, and the other displays "Thread2" every 2000 milliseconds. Create the threads by extending the Thread class	IV	2
26.	Write a program that executes two threads. One thread will print the even numbers and another thread will print odd numbers from 1 to 200.	IV	2
27.	*Write a program in Java to perform read and write operations on a Text file.	V	2
28.	Write a program in Java to demonstrate use of List. 1) Create ArrayList and add weekdays (in string form) 2) Create LinkedList and add months (in string form) Display both List.	V	2
29.	Write a program in Java to create a new HashSet, add colors (in string form) and iterate through all elements using for-each loop to display the collection.	V	2
30.	*Write a Java program to create a new HashMap, add 5 students' data (enrolment no and name). retrieve and display the student's name from HashMap using enrolment no.	V	2

Note

- i. More **Practical Exercises** can be designed and offered by the respective course teacher to develop the industry relevant skills/outcomes to match the COs. The above table is only a suggestive list.
- ii. The following are some **sample** 'Process' and 'Product' related skills (more may be added/deleted depending on the course) that occur in the above listed **Practical Exercises** of this course required which are embedded in the COs and ultimately the competency..

To learn Object-Oriented programming concepts and techniques using the Java programming language. To learn to write, test, and debug introductory level Object-Oriented programs using

Java. In addition, the student will be introduced to the following concepts, which are important workforce activities:

S. No.	Sample Performance Indicators for the PrOs	Weightage in %
1	Correctness of the program/application	30
2	Readability and documentation of the program/application	10
3	Design/Develop/Implement Program/application – Write code, integrate subsystems	20
4	Debugging ability – Perform unit testing, Test and validate program/application	20
5	Execution of the program/application and answer to the sample questions.	20
Total		100

6. MAJOR EQUIPMENT/ INSTRUMENTS AND SOFTWARE REQUIRED

These major equipment/instruments and Software required to develop PrOs are given below with broad specifications to facilitate procurement of them by the administrators/management of the institutes. This will ensure conduction of practical in all institutions across the state in proper way so that the desired skills are developed in students.

S. No.	Equipment Name with Broad Specifications	PrO. No.
1	Computer with latest configuration with Windows/Linux/Unix Operating System.	All
2	JDK (Java Development Kit) Version 8 or above	All
3	Any editor - Notepad++, Visual Studio Code, Eclipse IDE, NetBeans IDE	All

7. AFFECTIVE DOMAIN OUTCOMES

The following **sample** Affective Domain Outcomes (ADOs) are embedded in many of the above-mentioned COs and PrOs. More could be added to fulfil the development of this competency.

- Motivation and Attitude towards learning
- Learning Methodology and Communication styles
- Use of technology
- Work as a leader/a team member.
- Follow ethical practices.

The ADOs are best developed through the laboratory/field-based exercises. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

- i. 'Valuing Level' in 1st year
- ii. 'Organization Level' in 2nd year.
- iii. 'Characterization Level' in 3rd year.

8. UNDERPINNING THEORY

The major Underpinning Theory is formulated as given below and only higher level UOs of *Revised Bloom's taxonomy* are mentioned for development of the COs and competency in the students by the teachers. (Higher level UOs automatically includes lower level UOs in them). If required, more such higher level UOs could be included by the course teacher to focus on attainment of COs and competency.

Unit	Unit Outcomes (UOs)	Topics and Sub-topics
Unit –1: Introduction to Java Programming Language	1a. Describe java features and applications and environment setup of Java programming language. 1b. Install Java Components 1c. Write simple program using java programming language 1d. Describe data types, identifiers, constants and variables 1e. Write programs using arrays 1f. List types of operators 1g. Write simple java programs using decision and control structures.	1. Introduction to Java and Brief history, java features, java Applications 2. Java components: Java Virtual Machine (JVM), Java Runtime Environment (JRE), JDK (Java Development Kit). Importance of byte code and Garbage Collection 3. Java environment setup; Structure of java program; Compilation and execution of java program, , Comment Syntax 4. Primitive Data Types : byte, short, int, long, float, double, char, Boolean 5. Identifiers, Declarations of constants & variables, Type Conversion and Type Casting, Scope of variables 6. Arrays of Primitive Data Types, Types of Arrays : one-dimensional and two-dimensional array 7. Different Operators: Arithmetic, Bitwise, Rational, Logical, Assignment, Conditional, Ternary, Increment and Decrement 8. Decision & Control Statements: Selection Statement (if, if...else, switch), Loops (while, do-while, for), Jump Statements : break, continue, return

Unit– 2: Object Oriented Programming Concepts	2a. Differentiate between POP and OOP 2b. List object oriented programming concepts 2c. Develop simple java program using class. 2d. Use this and final keyword 2e. Write object oriented program using constructor 2f. Write java program using String class	1. Procedure-Oriented vs. Object-Oriented Programming concept 2. Basics of OOP: Class, Object, Encapsulation, Polymorphism Abstraction, Inheritance 3. Defining classes, fields and methods, creating objects 4. Accessing rules : public, private, protected, default 5. this keyword, static keyword, final keyword 6. Constructors: Default constructors, Parameterized constructors, Copy constructors, Passing object as a parameter 7. method overloading, constructor overloading 8. Wrapper Classes, String Class and its methods: charAt(), contains(), format(), length(), split() 9. User Input: Scanner class and Command Line Arguments
Unit– 3: Inheritance, Packages & Interfaces	3a. List types of inheritance 3b. Write program to implement single, multilevel, hierarchical inheritance 3c. Write programs to implement method overriding 3d. Write programs to implement overriding using Object class 3e. Write programs to implement multiple inheritance	1. Basics of Inheritance, Types of inheritance: single, multiple, multilevel, hierarchical and hybrid inheritance, 2. method overriding, Object class and overriding its methods : equals(), toString(), finalize(), hashCode() 3. Defining interface, implementing interface, multiple inheritance using interface 4. Abstract class and final class 5. Creating package, importing package, access rules for packages

	3f. Create a user-defined package and use that package	
Unit-4: Exception Handling & Multithreading	4a. Describe errors and types of exceptions 4b. List types of errors 4c. Write user-defined exceptions 4d. Define thread, creating threads, multithreading, thread priority & synchronization	1. Types of errors, exceptions, try...catch statement, multiple catch blocks, throw and throws keywords, finally clause, uses of exceptions, user defined exceptions 2. Concept of Multithreading, Creating thread, extending Thread class, implementing Runnable interface, life cycle of a thread, Thread priority, Thread exception handling in threads
Unit-5: File Handling and Collections Framework	5a. Describe basics of streams, stream classes, creation, reading and writing files in context to file handling 5b. Describe Collections framework 5c. Write programs using ArrayList and LinkedList 5d. Write programs to Map classes	1. Stream classes, class hierarchy, useful I/O classes : FileInputStream, FileOutputStream 2. Creation of text file, reading and writing text files 3. Collections Framework overview, Collection classes- ArrayList, LinkedList, HashSet. The For-Each loop 4. Map class : HashMap

Note: The UOs need to be formulated at the 'Application Level' and above of Revised Bloom's Taxonomy' to accelerate the attainment of the COs and the competency.

9. SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Introduction to Java Programming Language	08	4	4	6	14
II	Object Oriented Programming Concepts	11	3	6	9	18

III	Inheritance, Packages and Interfaces	11	3	6	9	18
IV	Exception Handling and Multithreading	06	2	3	5	10
V	File Handling and Collections Framework	06	2	3	5	10
Total		42	14	22	34	70

Legends: R=Remember, U=Understand, A=Apply and above (Revised Bloom's taxonomy)

Note: This specification table provides general guidelines to assist student for their learning and to teachers to teach and question paper designers/setters to formulate test items/questions assess the attainment of the UOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may vary from above table.

10. SUGGESTED STUDENT ACTIVITIES

Other than the classroom and laboratory learning, following are the suggested student-related **co-curricular** activities which can be undertaken to accelerate the attainment of the various outcomes in this course: Students should conduct following activities in group and prepare small reports (of 1 to 5 pages for each activity). For micro project report should be as per suggested format, for other activities students and teachers together can decide the format of the report. Students should also collect/record physical evidences such as photographs/videos of the activities for their (student's) portfolio which will be useful for their placement interviews:

- Undertake micro-projects in teams.
- List different versions of java and their new features
- Prepare charts to explain use/process of the identified topic.
- <https://java-iitd.vlabs.ac.in/> this website provides virtual lab and gives simulation of java programs.
- Students are encouraged to register themselves in various MOOCs such as: Swayam, edX, Coursera, Udemy etc. to further enhance their learning.
- <https://www.codechef.com> this website gives competitive programming problems, students are expected to solve examples and crosscheck with output.
- Encourage students to form a coding club at institute level and can help the slow learners

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES

These are sample strategies, which the teacher can use to accelerate the attainment of the various outcomes in this course:

- Massive open online courses (**MOOCs**) may be used to teach various topics/sub topics.
- Guide student(s) in undertaking micro-projects.
- Managing Learning Environment
- Diagnosing Essential Missed Learning concepts that will help for students.
- Guide Students to do Personalized learning so that students can understand the course material at his or her pace.
- Encourage students to do Group learning by sharing so that teaching can easily be enhanced.
- 'CI' in section No. 4** means different types of teaching methods that are to be employed by teachers to develop the outcomes.

- h) About **20% of the topics/sub-topics** which are relatively simpler or descriptive in nature is to be given to the students for **self-learning**, but to be assessed using different assessment methods.
- i) With respect to **section No.10**, teachers need to ensure to create opportunities and provisions for **co-curricular activities**.
- j) Guide students on how to address issues on environment and sustainability using the knowledge of this course

12. SUGGESTED MICRO-PROJECTS

Only one micro-project is planned to be undertaken by a student that needs to be assigned to him/her in the beginning of the semester. In the first four semesters, the micro-project are group-based (group of 3 to 5). However, **in the fifth and sixth semesters**, the number of students in the group should **not exceed three**.

The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. Each student will have to maintain dated work diary consisting of individual contribution in the project work and give a seminar presentation of it before submission. The total work load on each student due to the micro-project should be about **16 (sixteen) student engagement hours** (i.e., about one hour per week) during the course. The students ought to submit micro-project by the end of the semester (so that they develop the industry-oriented COs).

A suggestive list of micro-projects is given here. This should relate highly with competency of the course and the COs. Similar micro-projects could be added by the concerned course teacher:

1. Student Record Keeping System
2. Inventory Management System
3. Airline reservation system
4. Electricity billing system
5. Healthcare management system
6. Library management system
7. Bank management system
8. Medical management system
9. Quiz management system
10. Survey System
11. Stock management system
12. Tic Tac Toe Game
13. Chess Game
14. Battleship Console Game

13. SUGGESTED LEARNING RESOURCES

Sr. No.	Title of Book	Author	Publication with place, year and ISBN
1	Java: The Complete Reference, Twelfth Edition	Herbert Schildt	Tata McGraw Hill

2	Programming with Java 6th Edition	E Balagurusamy	Tata McGraw Hill
3	Sachin Malhotra &SaurabhChoudhary	Programming in JAVA, Second Edition	Oxford
4	Head First Java: A Brain-Friendly Guide, Third Edition	Kathy Sierra, Bert Bates, Trisha Gee	Shroff/O'Reilly

14. SUGGESTED LEARNING WEBSITES

- <https://docs.oracle.com/javase/tutorial/> (Documentation)
- <https://www.w3schools.com/java/default.asp> (Blog)
- <https://www.javatpoint.com/java-tutorial> (Blog)
- <https://www.geeksforgeeks.org/java/> (Blog)
- <https://cse.iitkgp.ac.in/~dsamanta/java/ch3.htm#Introduction> (Blog)
- <https://www.youtube.com/playlist?list=PLbRMhDVUMngcx5xHChJ-f7ofxZI4JzuQR> (Video Series)
- https://onlinecourses.nptel.ac.in/noc23_cs46/preview (Certification)
- <https://in.coursera.org/specializations/java-programming> (Certification)
- <https://www.edx.org/course/software-development-fundamentals> (Certification)
- https://www.w3schools.com/java/tryjava.asp?filename=demo_compiler (Online Editor)

15. PO-COMPETENCY-CO MAPPING

Semester II	Basics of Operating System (Course Code:)									
	POs and PSOs									
Competency & Course Outcomes	PO 1 Basic & Discipline specific knowledge	PO 2 Problem Analysis	PO 3 Design/ development of solutions	PO 4 Engineering Tools, Experimentation & Testing	PO 5 Engineering practices for society, sustainability & environment	PO 6 Project Management	PO 7 Life-long learning	PSO 1	PSO 2	PSO 3 (If needed)
Competency Develop java application using object-oriented approach.										
Course Outcomes CO a) Write simple java programs for a given problem statement.	2	-	-	-	-	-	2			
CO b) Use object oriented programming concepts to solve real world problems.	2	3	3	2	-	2	2			
CO c) Develop an object- oriented program using inheritance and package concepts for a given problem statement.	2	3	3	2	-	2	2			
CO d) Develop an object oriented program using	2	3	3	2	-	2	2			

multithreading and exception handling for a given problem statement.										
CO e) Develop an object-oriented program by using the files and collection framework.	3	3	3	2	-	2	2			

Legend: '3' for high, '2' for medium, '1' for low or '-' for the relevant correlation of each competency, CO, with PO/ PSO

16. COURSE CURRICULUM DEVELOPMENT COMMITTEE**GTU Resource Persons**

S. No.	Name and Designation	Institute	Contact No.	Email
1	Shri S. N. Sampat - HOD	LE College, Morbi	9033777389	snsampat@gmail.com
2	Smt. M. P. Mehta - HOD	Government Polytechnic, Himmatnagar	9879578273	manishamehtain@gmail.com
3	Smt. M. V. Prajapati - Lecturer	Government Polytechnic Gandhinagar	9428049861	mvprajapati2014@gmail.com
4	Shri M. J. Dabgar - Lecturer	Government Polytechnic Palanpur	8128576285	milav.dabgar@gujgov.edu.in