

GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)

Competency-focused Outcome-based Green Curriculum-2021 (COGC-2021)

Semester -IV

Course Title: Object Oriented Programming with JAVA

(Course Code: 4341602)

Diploma programme in which this course is offered	Semester in which offered
Information Technology	4 th Semester

1. RATIONALE

Java is a general-purpose computer programming language; it is platform independent, open-source, class-based, and object-oriented with enriched open source libraries. It is a simple, portable, distributive, robust, secure, dynamic, architecture neutral, object oriented programming language. It has idealized 'virtual machine' that allows application developers "write once, run anywhere", meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. In the current industrial scenario Java has the broad industry support and prerequisite to many technologies like Java Server Pages, and Android Application Development. This course develops necessary skills in students, after learning this course; students will be able to develop object oriented applications using java.

2. COMPETENCY

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

- **Develop a program using object oriented programming concepts with java to solve the given problems.**

3. COURSE OUTCOMES (COs)

The student will develop underpinning knowledge, adequate programming skills of competency for implementing various applications using object oriented programming with java to attain the following course outcomes.

- **Understand OOP (Object-Oriented Programming) concepts with java.**
- **Understand building blocks of OOPs language, inheritance, package and interfaces.**
- **Implement exception handling and multithreading in object oriented programs.**
- **Develop an object oriented program handling a Text file.**

3. TEACHING AND EXAMINATION SCHEME

Teaching Scheme (In Hours)			Total Credits (L+T/2+P/2)	Examination Scheme				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	CA	ESE	CA	ESE	
3	-	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: **L**-Lecture; **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) are the subcomponents of the COs. These PrOs need to be attained to achieve the COs.

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. required
1	Install JDK and Setup a Java Programming development environment by using: 1) Command Prompt (SET PATH command and using Environment Variable). 2) Any open source IDE (Eclipse, Jcreator etc)	I	01
2	Test the java development environment setup by implementing a simple java program (print: "OOP with JAVA").	I	01
3	Develop a basic java program that demonstrates data types of JAVA.	I	01
4	Develop a Java program to swap two numbers without using a temporary variable and with using a temporary variable (use command line argument to accept value from user).	I	01
5	Develop programs to demonstrate use of - 1) if statement and its different form 2) switch case statement	I	01
6	Develop program to demonstrate use of- 1) for loop 2) 'while' and 'do while' loop	I	01
7	Develop a Java program to find maximum and minimum numbers from array elements.	I	01
8	Develop a basic java program that demonstrates the use of Class & Object.	II	01
9	Develop a java program to find the factorial of a given number using a recursive function.	II	02
10	Develop a java program that demonstrates method overloading.	II	02
11	Develop a program for implementation of different functions of String class.	II	02
12	Develop a program for implementation of Wrapper Class to convert primitive value into object (Boxing) and object into primitive value (Un-boxing).	II	02

13	Develop a program with a static block and show that it will be executed before the main () method in a class.	II	01
14	Develop a program to demonstrate use of static functions.	II	01
15	Develop a program to demonstrate use of 'this' keyword. Check whether 'this' can access the private members of the class or not.	II	02
16	Develop a program with an overloaded constructor. Also develop the copy constructor to create a new object with the state of the existing object.	II	02
17	Develop a program to demonstrate the use of private constructor and also write a method which will count the number of instances created using default constructor only.	II	02
18	Develop a program to demonstrate single inheritance, multilevel inheritance, and hierarchical inheritance.	III	02
19	Develop a program with one class named shape which has two member functions named erase () and draw (). In the program we have three other subclasses: circle, triangle and square. override methods of the superclass into subclasses.	III	02
20	Develop a program for implementation of Dynamic method dispatch.	III	02
21	Develop a java program that demonstrates the use of Abstract class.	III	02
22	Develop a java program that illustrates interface inheritance. Interface 'A1' and 'A2' are extended from interface 'A'. Interface 'A12' inherited from both 'A1' and 'A2'. Each interface declares one method and one constant. Class 'Interface_Imple' implements 'A12'. Instantiate 'Interface_Imple' and invoke each of its methods. Each method displays one of the constants.	III	02
23	Develop a program to create a Package and demonstrate how packages are used in java. And use java access modifier to demonstrate the access rules in a package.	III	02
24	Develop a program to demonstrate the use of 'super' and 'final' keywords.	III	02
25	Develop programs to demonstrate the use of Exception Handling using predefined Exception Classes.	IV	02
26	Develop a program to handle multiple exceptions using multiple try blocks and multiple catch blocks.	IV	02
27	Develop a program to implement user defined exceptions.	IV	02
28	Develop a program to demonstrate use of throw, throws, and finally keyword.	IV	02

29	Develop a program that executes two threads. One thread displays "Java Programming" every 2 seconds, and the other displays "Semester - 4th" every 5 seconds.(Create the threads by extending the Thread class)	IV	02
30	Develop a program that executes two threads. One thread will print the even numbers and the another thread will print odd numbers between 1 to 10.(Create the thread by implementing runnable interface)	IV	02
31	Develop a program to demonstrate use of synchronization of threads when multiple threads are trying to update a common variable.	IV	02
32	Develop programs to create, write, modify, read operations on Text files.	V	04
	Total		56

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..

S. No.	Sample Performance Indicators for the PrOs	Weightage in %
1	Representation of classes and objects.	20
2	Coding methodology.	30
3	Testing and Debugging of the program.	20
4	Correctness of Program.	20
5	Submission in time.	10
	Total	100

6. MAJOR EQUIPMENT/ INSTRUMENTS/SOFTWARES REQUIRED

This major equipment/instrument/software with broad specifications for the PrOs is a guide to procure them by the administrators. This will ensure conduction of practical's in all institutions across the state in the proper way so that the desired skills are developed in students.

S. No.	Equipment Name with Broad Specifications	PrO. No.
1	Computer system with latest configuration.	All
2	JDK 1.8 (Java Development Kit) or above. Text-Editor: Notepad or any other. Open Source IDE: Eclipse, IntelliJ IDEA, Jcreator or any other.	

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 fulfill the development of this competency.

- a) Work as a leader/a team member.
- b) 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

Only the major Underpinning Theory is formulated as higher-level UOs of *Revised Bloom's taxonomy* in order development of the COs and competency is not missed out by the students and teachers. If required, more such higher-level UOs could be included by the course teacher to focus on the attainment of COs and competency.

Unit	Unit Outcomes (UOs)	Topics and Sub-topics
Unit – I Introduction to Object Oriented Programming paradigm.	1.1. Describe Programming Paradigm. 1.2. Differentiate between POP and OOP. 1.3. Understand important OOP fundamentals. 1.4. Understand basics of java programming and environment setup of Java.	1.1.1 Introduction to programming paradigm, types of programming paradigm. 1.2.1 Procedure-Oriented vs. Object-Oriented Programming concept. 1.3.1 Basics of OOP: Class, Object, data abstraction, encapsulation, inheritance, benefits of inheritance, polymorphism. 1.4.1 Basics of Java, Background/History of Java, Java and the Internet, Advantages of Java. 1.4.2 JDK, JRE, JVM, and Byte code. 1.4.3 Java Environment Setup. 1.4.4 Java program structure. 1.4.5 Compiling and running a simple java program. 1.4.6 Comments in java, data types, variables, scope and lifetime of variables, operators, type conversion and casting, Control Statements – If, else, nested if, if-else ladders, Switch, while, do-while, for, for-each, break, continue, arrays in java. 1.4.7 Command line argument 1.4.8 Garbage Collection.

<p>Unit – II</p> <p>Object Oriented Programming Concepts.</p>	<p>2.1. Define Objects and Classes and methods.</p> <p>2.2. Differentiate String and StringBuffer class.</p> <p>2.3. Explain Constructor with its types.</p>	<p>2.1.1 Defining classes, creating objects and methods, Passing and Returning object form Method, Method overloading.</p> <p>2.2.1 String class, StringBuffer class, Operations on string, StringJoiner class, Wrapper Class.</p> <p>2.2.2 Access control, modifiers, this keyword, static keyword.</p> <p>2.3.1 Constructors: Default constructors, Parameterized constructors, Copy constructors, Private constructor, and Constructor Overloading.</p>
<p>Unit-III</p> <p>Inheritance, Interface and Package.</p>	<p>3.1. Explain Inheritance with its types.</p> <p>3.2. Describe abstract class and Interface.</p> <p>3.3. Describe Creating package, importing package, access rules for packages, class hiding rules in a package.</p>	<p>3.1.1 Basics of Inheritance, Types of Inheritance, Method overriding, super and final keyword.</p> <p>3.1.2 Basics of Polymorphism, Types of Polymorphism, Difference between method overloading and method overriding.</p> <p>3.1.3 Dynamic method dispatch & Object class.</p> <p>3.2.1 Abstract classes v/s Interfaces, defining an interface, implementing interfaces, extending interfaces, default method, lambda expression.</p> <p>3.3.1 Creating package, setting a CLASSPATH, adding class and interfaces to a package, importing package, static import</p> <p>3.3.2 Java access modifier, Access and class hiding rules in a package.</p>

Unit-IV Exception Handling & Multithreaded Programming.	4.1. Explain exceptions with its type. 4.2. Define thread, creating threads, setting up thread priority & thread synchronization.	4.1.1 Fundamentals of Exception and Errors, Types of Exception. 4.1.2 Using try and catch in Exception, Multiple catch clauses, Use of nested try statements. 4.1.3 Throw and throws keywords, and finally clause. 4.1.4 Built in exceptions, creating own exception subclasses, Java Optional class. 4.2.1 Basics of Multithreading, The Java thread model and main thread, Creation of thread by extending Thread class, implementing Runnable interface. 4.2.2 Life cycle of a thread. 4.2.3 Thread priorities, Thread synchronization, inter thread communication, alive () & join () in thread. 4.2.4 Exception handling in threads
Unit-V File handling in Java.	5.1. Explain basics of streams, stream classes, creation, reading and writing files in context to file handling.	5.1.1 Introduction to Stream, types of Stream. 5.1.2 Stream classes and its hierarchy, I/O classes: File Class, File InputStream, File Output Stream, InputStreamReader, OutputStreamWriter, FileReader, FileWriter, Buffered Reader. 5.1.3 Reading and writing text files.

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	Total Marks
I	Introduction to Object Oriented Programming paradigm	10	06	06	04	16
II	Object Oriented Programming Concepts	06	04	04	04	12
III	Inheritance, Interface and Package.	10	04	06	06	16
IV	Exception Handling & Multithreaded Programming.	10	04	06	06	16
V	File handling in Java.	06	00	06	04	10
Total		42	18	28	24	70

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

Note: This specification table provides general guidelines to assist students 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 slightly from the 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 reports of about 5 pages for each activity, also collect/record physical evidences for their (student's) portfolio which will be useful for their placement interviews:

- a) Prepare a report based on practical performed in the laboratory.
- b) Undertake micro-projects in teams
- c) Give a seminar on any relevant topics.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

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

- a) Massive open online courses (**MOOCs**) may be used to teach various topics/subtopics.
- b) Guide student(s) in undertaking micro-projects.
- c) '**L**' in **section No. 4** means different types of teaching methods that is to be employed by teachers to develop the outcomes.
- d) 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.
- e) With respect to **section No.10**, teachers need to ensure to create opportunities and provisions for **co-curricular activities**.
- f) Guide students for open source editors.

13. 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 is group-based. However, in the fifth and sixth semesters, it should be preferably be **individually** undertaken to build up the skill and confidence in every student to become problem solver so that s/he contributes to the projects of the industry. In special situations where groups have to be formed for micro-projects, 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 a dated work diary consisting of individual contributions in the project work and give a seminar presentation of it before submission. The total duration of the micro-project should not be less than **16 (sixteen) student engagement hours** during the course. The

student ought to submit a micro-project by the end of the semester to develop the industry oriented COs.

A suggestive list of micro-projects is given here. This has to match the competency and the COs. Similar micro-projects could be added by the concerned course teacher:

Case Study 1: Develop a Banking management application that provides following services to customers.

1. Customers can view their account details such as type of account, available balance etc.
2. He/she can perform transactions (i. e. Cash Deposit and Withdraw).

Case Study 2: Develop a simple college management application that provides following services.

1. Stores Details of college, departments, and students (use inheritance to define class hierarchy).
2. Shows exam results of Students (Result of Individual student, %pass, %fail, top-3 students, etc).

Case Study 3: Develop a library book issue management system.

14. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication with place, year and ISBN
1	Java: The Complete Reference	Herbert Schildt	McGraw Hill Education, 11 th Edition ISBN-13 : 1260440232-978
2	Programming with Java	Balagurusamy E.	McGraw Hill Education, 5 th Edition ISBN-13:978-93-5134-320-2
3	Java 8 Programming Black Book	DT Editorial Services	Dreamtech Press, New Delhi, ISBN:978-93-5119-758-4

15. SOFTWARE/LEARNING WEBSITES

1. Java Development Kit:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. <https://docs.oracle.com/javase/tutorial/java/index.html>
3. <https://nptel.ac.in/courses/106105191>
4. <https://cse.iitkgp.ac.in/~dsamanta/java/index.htm>
5. <https://www.javatpoint.com/java-tutorial>
6. <https://www.tutorialspoint.com/java/index.htm>
7. <https://www.programiz.com/java-programming>
8. <https://www.geeksforgeeks.org/java/>
9. <https://www.guru99.com/java-tutorial.html>

16. PO-COMPETENCY-CO MAPPING

Semester II	Object Oriented Programming with JAVA(Course Code: 4341602)						
	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 And Testing	PO 5 Engineering practices for society, sustainability & environment	PO 6 Project Management	PO 7 Life-long learning
Competency							
Develop a program using object oriented programming concepts with java to solve the given problem.							
Course Outcomes							
Understand OOP (Object-Oriented Programming) concepts with java.	3	2	3	2	2	-	3
Understand building blocks of OOPs language, inheritance, package and interfaces.	3	2	3	3	2	2	3
Implement exception handling and multithreading in object oriented programs.	3	3	3	3	2	2	3
Develop an object oriented program handling Text and JSON file.	2	3	3	3	2	2	3

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

17. COURSE CURRICULUM DEVELOPMENT COMMITTEE

GTU Resource Persons

S. No.	Name and Designation	Institute	Email
1	Ruchik N. Jani Lecturer in I.T.	L. E. College (Polytechnic), Morbi.	janiruchik@gmail.com
2	Shilpa K. Rathod Lecturer in I.T.	L. E. College (Polytechnic), Morbi.	shilu.avi418@gmail.com
3	Suryadeepsinh P. Jadeja Lecturer in I.T.	L. E. College (Polytechnic), Morbi.	suryadeep2309@gmail.com
4	Ajaypalsinh. S. Gohil Lecturer in I.T.	Sir B. P. T. I, Bhavnagar	ajaypal.gohil@gmail.com