

Annexure III: Teaching Strategy

| CLO No. | Teaching-Learning Activities for Active Learning | Assessment (Formative and Summative) Activities |
|----------------|---|--|
| CLO 1 | 1. Discussion 2. Problem Solving 3. Assignment on topics 4. Practical sessions | 1. Continuous Assessment 2. Practical exam 3. Assignment |
| CLO 2 | 1. Discussion 2. Problem Solving 3. Assignment on topics 4. Practical Sessions | 1. Continuous Assessment 2. Practical exam 3. Assignment |
| CLO 3 | 1. Problem Solving 2. Assignment on topics 3. Practical Sessions | 1. Continuous Assessment 2. Practical exam 3. Assignment |
| CLO 4 | 1. Problem Solving 2. Assignment on topics 3. Practical Sessions | 1. Continuous Assessment 2. Practical exam 3. Assignment |

Annexure IV: Course Outline

| | |
|---|---|
| Institute | Institute of Technology |
| Department | Computer Science and Engineering |
| Programme | B.Tech. |
| Batch | 2024-25 |
| Term/Semester | 3 rd |
| Course Title | Object Oriented Programming |
| Course Code | 2CS502 |
| Credit Hours | L-2+P-2 |
| Faculty's Name | Prof Kruti Lavingia |
| E-mail ID | kruti.lavingia@nirmauni.ac.in |
| Contact No. | 079716529572 |
| Office Hours | Mon-Sat: 8:45 am to 4:00 pm Sat: 9:00 am to 3:00 pm |
| Moodle Details (as and when available) | https://lms.nirmauni.ac.in/course/view.php?id=1016 |

a. Course Overview

Object-oriented programming (OOP) is a programming paradigm based on the concept of objects, which are data structures that contain data, in the form of fields (or attributes) and code, in the form of procedures, (or methods). The focus of procedural programming is to break down a programming task into a collection of variables, data structures, and subroutines, whereas in object-oriented programming it is to break down a programming task into objects that expose behavior (methods) and data (fields) using interfaces. The most important distinction is that while procedural programming uses procedures to operate on data structures, object-oriented programming bundles the two together, so an object, which is an instance of a class, operates on its “own” data structure.

An object can be defined as a data field that has unique attributes and behavior. object-oriented programming focuses on the objects that developers want to manipulate rather

than the logic required to manipulate them. This approach to programming is well-suited for programs that are large, complex and actively updated or maintained. This includes programs for manufacturing and design, as well as mobile applications; for example, object-oriented programming can be used for manufacturing system simulation software.

The organization of an object-oriented program also makes the method beneficial to collaborative development, where projects are divided into groups. Additional benefits of object-oriented programming include code reusability, scalability and efficiency.

The first step in object-oriented programming is to collect all of the objects a programmer wants to manipulate and identify how they relate to each other -- an exercise known as data modelling. Examples of an object can range from physical entities, such as a human being who is described by properties like name and address, to small computer programs, such as widgets. Once an object is known, it is labelled with a class of objects that defines the kind of data it contains and any logic sequences that can manipulate it. Each distinct logic sequence is known as a method. Objects can communicate with well-defined interfaces called messages.

b. Course Learning Outcomes (CLOs) [CLO sequence can be reorder]

At the end of the course, the students shall be able to:

1. explain procedural and object-oriented paradigms and principles **(BL-2)**
2. relate the concepts of object-oriented design with principles of object-oriented programming **(BL-2)**
3. apply exception handling, input-output operations and multi-threading concepts for application development **(BL-3)**
4. make use of object-oriented concepts and design for developing various programs **(BL-3)**

c. Text Book (reflect other study/reference materials in session plan)

1. Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill
2. Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill
3. David Flanagan, Student Workbook Java in a Nutshell O'Reilly
4. Cay S. Horstmann Core Java(TM), Volume I—Fundamentals Prentice Hall

d. Assessment Components & Schedule

| SEE | CE | LPW |
|-----|-----|-----|
| 0.4 | 0.3 | 0.3 |

| Component (with No. of sub-components) | Weightage (each part of the Component) | Schedule | Marks | CLO Number |
|--|--|--------------------------|-------|--------------|
| Quiz | 20% of CE | Will be updated soon | 20 | 1,2,3, and 4 |
| Sessional Test | 60% of CE | 09/10/23 to 16/10/23 | 50 | 1,2,3, and 4 |
| Term Paper/Assignments | 20% of CE | Will be updated soon | 50 | 1,2,3, and 4 |
| SEE | 100% of SEE | 14/12/23 Onwards | 100 | 1,2,3 and 4 |
| LPW | 100% of LPW | 04/12/2023 to 08/12/2023 | 100 | 1,2,3 and 4 |

e. Session Plan

| Sess. No. | Topic/Sub Topic | Session Details | |
|-----------|---|---------------------------------------|---|
| 1 | Introduction to Course and Course Policy presentation A Review of programming paradigms, Introduction to Object Oriented Programming | Text Book | --- |
| | | Reading (Reference) | The Complete Reference- JAVA by Herbert Schildt (7th Ed.) 1. https://www.javatpoint.com/java-tutorial 2. https://www.tutorialspoint.com/java/index.htm |
| | | Pedagogy | Open discussion |
| | | Course learning outcome (CLO) | 1,2,3,4 |
| | | Session Learning Outcome (SLO) | Understand importance, scope and policy of the course |

| | | | |
|---|--|----------------------------|---|
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 2 | Comparison of Object Oriented approach with other programming approaches | Text Book | --- |
| | | Reading (Reference) | <ul style="list-style-type: none"> • The Complete Reference- JAVA by Herbert Schildt • https://www.javatpoint.com/java-tutorial • https://www.tutorialspoint.com/java/index.htm |
| | | Pedagogy | Power Point Presentation (PPT) |
| | | CLO | 1 |
| | | SLO | <ul style="list-style-type: none"> • Understand the significance of Objects and classes and the concept of bottup up approach in programming |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 3 | History and Overview of Java: Creation of Java, , Evolution of Java, features of Java, byte code, Java Development Kit (JDK), Java Virtual Machine (JVM) | Text Book | --- |
| | | Reading (Reference) | <ul style="list-style-type: none"> • The Complete Reference- JAVA by Herbert Schildt (7th Ed.) 3-14 • https://www.javatpoint.com/history-of-java |
| | | Pedagogy | Open discussion, Power Point Presentation (PPT) |
| | | CLO | 1 |
| | | SLO | Understand features of Java, byte code, Java Development Kit (JDK), Java Virtual Machine (JVM) |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 4 | Introduction to three OOP principles (Inheritance, Polymorphism, Encapsulation), Introduction to classes and methods | Text Book | --- |
| | | Reading (Reference) | The Complete Reference- JAVA by Herbert Schildt (7th Ed.) 15-21 |
| | | Pedagogy | Power Point Presentation (PPT), Demonstration of program |
| | | CLO | 2 |
| | | SLO | Understand the basic principles of OOP and basics of Classes in Java |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 5 | | Text Book | --- |

| | | | |
|---|---|----------------------------|---|
| | Data types, variables and Operators: primitive data types, literals, variables, type conversion and casting | Reading (Reference) | The Complete Reference- JAVA by Herbert Schildt (7th Ed.) 21-32 |
| | | Pedagogy | Power Point Presentation (PPT), Demonstration of program |
| | | CLO | 2 |
| | | SLO | <ul style="list-style-type: none"> Know how to use data types, variables in Java Understand concepts of type casting and type conversion and apply those in programming |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 6 | Automatic type promotion in expressions, type promotion rules, Basic of Array and String | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill (Pg. No. 47-50) Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill |
| | | Pedagogy | PowerPoint Presentation (PPT), Notepad (run java code using command prompt), Chalk and board talk, |
| | | CLO | 1 |
| | | SLO | Understand the concept of array and string in java with suitable java programs. Brief overview of automatic type promotion in expressions and type promotion rule |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 7 | Arithmetic operators, bitwise operators | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill (Chapter 4, Pg. No. 57-70) Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill |
| | | Pedagogy | Notepad (run java code using command prompt) , PowerPoint Presentation (PPT) |
| | | CLO | 1 |
| | | SLO | Understanding and problem solving based on arithmetic and bitwise operators through a suitable java programs |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |

| | | | |
|----|--|---------------------------|--|
| 8 | Relational operators, Boolean logical operators, Assignment operators, ternary operators, operator precedence | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill (Chapter 4, Pg. No. 70-75) Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill |
| | | Pedagogy | PowerPoint Presentation (PPT) |
| | | CLO | 1 |
| | | SLO | Different application and example solving based on the different types of operators, Precedence and associativity of operators |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 9 | Control Statements: Selection statements : if, switch Iteration statements : while, do-while, for, Iteration statements : the for-each version of the for Loop, Nested Loops | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill (Chapter 5, Pg. No. 77-103) Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill |
| | | Pedagogy | Power Point Presentation (PPT) |
| | | CLO | 1,2 |
| | | SLO | Understanding the various conditional constructs and looping constructs in java with simple programs. |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 10 | Jump statements: break and continue Arrays: one dimensional array, multi-dimensional array, alternative array declaration statements | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill (Chapter 3, Pg. No. 48-55) Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill |
| | | Pedagogy | Chalk and board talk, Power point presentation, Power Point Presentation (PPT), Notepad (run java code using command prompt) |
| | | CLO | 1,4 |

| | | | |
|----|--|----------------------------|--|
| | | SLO | Understand the user defined data type, different array types, overview of simple programs based on multi-dimensional array |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 11 | Classes and Methods: class fundamentals, declaring objects, assigning object reference variables Introduction to method: Adding a method to the class, returning a value, Adding a Method that takes parameters | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java2 — The Complete Reference, Tata McGraw Hill. (Chapter6, page no.105 to 115.) Balaguruswamy, Programming with Java — A primer, Tata McGraw Hill |
| | | Pedagogy | Chalk and board talk, Powerpoint presentation |
| | | CLO | 1,4 |
| | | SLO | Understand the Classes and Methods: class fundamentals, declaring objects, assigning object reference variables, Adding a method to the class. |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 12 | Constructors, Parameterized constructor, Garbage collection, Overloading methods, overloading constructors | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java2 — The Complete Reference, Tata McGraw Hill. (Chapter6, page no.115 to 128.) Balaguruswamy, Programming with Java — A primer, Tata McGraw Hill |
| | | Pedagogy | Chalk and board talk, Powerpoint presentation Notepad (run java code using command prompt) |
| | | CLO | 1,4 |
| | | SLO | Understand the Constructors, Parameterized constructor, Garbage collection, Overloading methods, overloading constructors. |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 13 | this keyword, finalize () method, Object as a parameter, Returning objects, Recursion. | Text Book | - |
| | | Reading (Reference) | Herbert Schildt, Java2 — The Complete Reference, Tata McGraw Hill. (Chapter6, page no.120 to 122, Chapter7, page no.125 to 135.) Balaguruswamy, Programming with Java — A primer, Tata McGraw Hill |
| | | Pedagogy | Chalk and board talk, Powerpoint presentation Notepad (run java code using command prompt) |
| | | CLO | 2 |

| | | | |
|----|---|---------------------------|--|
| | | SLO | Understand this keyword, finalize() method, Object as a parameter, Returning objects, Recursion. |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 14 | Access control: static, final, Nested and inner classes, Command line arguments, variable-length arguments | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java2 — The Complete Reference, Tata McGraw Hill. (Chapter7, page no.138 to 150.) Balaguruswamy, Programming with Java — A primer, Tata McGraw Hill |
| | | Pedagogy | Chalk and board talk, Powerpoint presentation Notepad (run java code using command prompt) |
| | | CLO | 2 |
| | | SLO | Understand the Access control, Command line arguments, variable-length arguments. |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 15 | String Handling: String constructors, String Operations | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java2 — The Complete Reference, Tata McGraw Hill. (Chapter7, page no.138 to 150.) Balaguruswamy, Programming with Java — A primer, Tata McGraw Hill |
| | | Pedagogy | Chalk and board talk, Powerpoint presentation Notepad (run java code using command prompt) |
| | | CLO | 1,2 |
| | | SLO | Understand the String Handling: String constructors, String Operations. |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 16 | String Methods | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java: The Complete Reference, Tata McGraw Hill |
| | | Pedagogy | Chalk and board lecture, Power Point Presentation (PPT) |
| | | CLO | 2 |
| | | SLO | Understand the Various String Methods in Java |

| | | | |
|----|---|---------------------------|---|
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 17 | String Buffer Class | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java: The Complete Reference, Tata McGraw Hill |
| | | Pedagogy | Power point presentation, Chalk and board talk. |
| | | CLO | 1, 2 |
| | | SLO | Understand the String Buffer Class constructor and methods |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 18 | Inheritances: Basics, member access and inheritance, Super class references, Using super | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java: The Complete Reference, Tata McGraw Hill |
| | | Pedagogy | chalk and board, Power point presentation |
| | | CLO | 2 |
| | | SLO | Understand the basics of Inheritance, how to access members of a class, and Super class variable reference to subclass |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 19 | Multilevel hierarchy, constructor call sequence, method overriding, Dynamic method dispatch, abstract classes, Object class | Text Book | |
| | | Reading(Reference) | Herbert Schildt, Java: The Complete Reference, Tata McGraw Hill |
| | | Pedagogy | Power point presentation |
| | | CLO | 2 |
| | | SLO | Understand the constructor chaining, Method overloading and overriding, Dynamic method dispatch (Run time polymorphism in Java) and Abstract classes and methods. |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 20 | Packages and Interfaces: defining a package, finding packages and CLASSPATH, access protection | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java: The Complete Reference, Tata McGraw Hill |
| | | Pedagogy | Chalk and board, Power point presentation. |
| | | CLO | 2 |
| | | SLO | Understand the concept of packages and Interface |

| | | | |
|----|---|---------------------------|---|
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 21 | Importing packages, interfaces (defining, implementation, nesting, applying), variables in interfaces, extending interfaces, instance of operator | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java: The Complete Reference, Tata McGraw Hill |
| | | Pedagogy | Power point presentation. |
| | | CLO | 3 |
| | | SLO | know concepts about Importing packages, interfaces (defining, implementation, nesting, applying), variables in interfaces, extending interfaces, instance of operator in Java |
| | | URL | https://lms.nirmauni.ac.in/login/index.php |
| 22 | Exception Handling: fundamental, exception types, uncaught exceptions, try, catch, | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java: The Complete Reference, Tata McGraw Hill |
| | | Pedagogy | Power point presentation. |
| | | CLO | 3 |
| | | SLO | understand and apply concepts of Exception Handling: fundamental, exception types, uncaught exceptions, try, catch, in Java |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 23 | throw, throws, finally, multiple catch clauses, nested try statements | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java: The Complete Reference, Tata McGraw Hill |
| | | Pedagogy | Power point presentation. |
| | | CLO | 3 |
| | | SLO | understand and apply concepts of throw, throws, finally, in Java |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 24 | built-in exceptions, custom exceptions (creating your own exception sub classes) | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java: The Complete Reference, Tata McGraw Hill |
| | | Pedagogy | Power point presentation. |
| | | CLO | 3 |

| | | | |
|----|--|---------------------------|--|
| | | SLO | create built-in exceptions, custom exceptions (creating your own exception sub classes) in Java |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 25 | Multithreaded Programming: Basics of Multithreading, Java thread model | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java: The Complete Reference, Tata McGraw Hill |
| | | Pedagogy | Power point presentation. |
| | | CLO | 3 |
| | | SLO | understand and apply concepts Multithreaded Programming: Basics of Multithreading, Java thread model |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 26 | Thread priorities, synchronization, messaging, Thread class | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill (Chapter 11, Pg. No. 223-254) Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill |
| | | Pedagogy | Notepad (run Java code using command prompt), PowerPoint Presentation (PPT), Chalk and board |
| | | CLO | 1,2,3 |
| | | SLO | Using suitable Java programs and concepts, students will learn how to set thread priorities, messaging, thread class and various thread synchronization. |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 27 | Runnable interfaces, creating a thread(s), Thread | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill (Chapter 11, Pg. No. 223-254) Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill |

| | | | |
|----|--|---------------------------|--|
| | class methods | Pedagogy | Notepad (run Java code using command prompt), PowerPoint Presentation (PPT), Chalk and board |
| | | CLO | 2,3,4 |
| | | SLO | How to work with a runnable interface, Also creating various thread and thread class methods using Java program and methods |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 28 | Synchronization, Interthread Communication | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill (Chapter 11, Pg. No. 223-254) Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill |
| | | Pedagogy | Notepad (run Java code using command prompt), PowerPoint Presentation (PPT), Chalk and board |
| | | CLO | 3,4 |
| | | SLO | Understand the thread synchronization and interthread communication. Implement with java code. |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 29 | Managing I/O: Streams, Byte Streams and Character Streams, Predefine | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill (Chapter 11, Pg. No. 223-254) Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill |
| | | Pedagogy | Notepad (run Java code using command prompt), PowerPoint Presentation (PPT), Chalk and board |

| | | | |
|----|---|---------------------------|--|
| | d Streams, Reading console Input | CLO | 3,4 |
| | | SLO | Learn about managing I/O. Understand difference between Streams, Byte Streams and Character Streams, Predefined Streams |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |
| 30 | Writing Console Output PrintWriter class, File management class | Text Book | - |
| | | Reading(Reference) | Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill (Chapter 11, Pg. No. 223-254) Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill |
| | | Pedagogy | Notepad (run Java code using command prompt), PowerPoint Presentation (PPT), Chalk and board |
| | | CLO | 2,3,4 |
| | | SLO | Introduce file management class in Java. Understand the printWriter class and implement using java code. |
| | | URL | https://lms.nirmauni.ac.in/course/view.php?id=1016 |

f. References

1. Herbert Schildt, Java – The Complete Reference, Tata McGraw Hill
2. Balaguruswamy, Programming with Java – A primer, Tata McGraw Hill
3. David Flanagan, Student Workbook Java in a Nutshell O'Reilly
4. Cay S. Horstmann Core Java(TM), Volume I—Fundamentals Prentice Hall

Some More References:

1. <https://www.javatpoint.com/java-tutorial>
2. <https://www.tutorialspoint.com/java/index.htm>
3. <https://docs.oracle.com/javase/tutorial/>

4. <https://www.geeksforgeeks.org/java/>

g. Instructions

Students are expected to carry out assigned work under Continuous Evaluation (CE) component and LPW component independently. Copying in any form is not acceptable and will invite strict disciplinary action. Evaluation of corresponding component will be affected proportionately in such cases. Turnitin software will be used to check plagiarism wherever applicable. Academic integrity is expected from students in all components of course assessment.



CERTIFICATE

This is to certify that

Mr. / Ms. _____

Roll No. _____ *of Semester* _____

has satisfactorily completed the course in

Within the four walls of the
Institute.

Date of Submission: _____

Sign In-charge

Head of the Department

List of Experiments

| Sr No. | Learning Objective | No. of Hours | Mapped CLO |
|--------|---|--------------|------------|
| 1 | <p>(Preliminaries and Basic Java Programming)</p> <p>a. Hands-on practice on C-Programming:</p> <p>i. Write a program in C to display and count a total number of duplicate elements in an array. Input the number of elements to be stored in the array :5 Input 5 elements in the array: element - 0: 5 element - 1: 1 element - 2: 1 element - 3: 2 element - 4: 2</p> <p>Expected Output: Total number of duplicate elements found in the array is: 2 Duplicate elements:1,2</p> <p>ii. Write a program in C to count the frequency of each element of an array. Test Data: Input the number of elements to be stored in the array :3 Input 3 elements in the array: element - 0: 25 element - 1: 12 element - 2: 43 Expected Output: The frequency of all elements of an array: 25 occurs 1 time 12 occurs 1 time 43 occurs 1 time</p> <p>b) Write a Java program to display greeting message like: “First Java Program....” on console.</p> <p>c) Write a Java program to display all primitive type variables. Also display your name in the last line.</p> | 02 | |

| | | | |
|---|---|----|--|
| 2 | <p>(Operators)</p> <p>a. Write a Java Program that check whether user entered number is special number or not. For example, Consider the number is 59. First, find the sum of all digits (5+9=14). Second, find multiplication of all digits (5*9=45). Then find addition of sum and multiplication of all digits (14+45=59). If it is same as number itself, than it is a special number.</p> <p>b. Write a Java program that demonstrate the concepts of automatic and explicit type casting.</p> <p>c. Write a Java program to:</p> <ol style="list-style-type: none"> check whether a number is odd or even (using if – else statement) check the category of a given character. (Using if...else...if ladder) check whether a number is prime or not. (Using for loop) <p>display reverse of a number and check whether it is palindrome or not. (Using while/do while loop)</p> <p>d. Pattern printing. (Using nested loops)</p> <pre> 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5 1 2 3 4 5 6 </pre> | 04 | |
| 3 | <p>(Operators and Array)</p> <p>Write a Java program for the following:</p> <p>a. Design calculator which contains arithmetic & bitwise operators. Operand(s) and operator must be scan from the user.</p> <p>b. Given an array of positive and negative numbers. Find if there is a subarray with 0 sum.</p> <p>Example: Input: 5 4 2 -3 1 6 Output: Yes Explanation: 2, -3, 1 is the subarray with sum 0.</p> | 04 | |
| 4 | <p>(Class and Objects)</p> <p>a. Create a class called complex for performing arithmetic operations with complex numbers. Use floating point variables to represent the private data of the class. Provide a default constructor that initialises the object with some default values. Provide public member methods for each of the following:</p> <ul style="list-style-type: none"> Addition of two complex numbers: It returns the result obtained by adding the respective real parts and the imaginary parts of the two complex numbers. The method must return complex class object. Subtraction of two complex numbers: It returns the result obtained by subtracting the respective real parts and the imaginary parts of | 02 | |

| | | | |
|---|--|----|--|
| | <p>the two complex numbers. The method must return complex class object.</p> <ul style="list-style-type: none"> display () – It displays the complex number in a+bi format. The output should be displayed as follows: - Sum of a_1+b_1i & a_2+b_2i is: a_3+b_3i | | |
| 5 | <p>(String Handling)</p> <p>a. Write a Java Program which asks user to enter a paragraph and perform the following operations:</p> <ol style="list-style-type: none"> Find total no of sentences in the paragraph and the total number of words in each sentence. Find the total number of characters in the entire paragraph and find out the occurrence of each character in the paragraph and display the information in proper format. Search a word (entered by the user) in the paragraph and print the position of the word (if found) or print appropriate message. <p>b. Implement a java program for scenario as given below: Write a program which takes a string (maximum 80 characters terminated by a full stop. The words in this string are assumed to be separated by one or more blanks. Arrange the words of the input string in descending order of their lengths. Same length words should be sorted alphabetically. Each word must start with an uppercase letter and the sentence should be terminated by a full stop. In the end store the final output in a text file.</p> <p>Test your program for the following data and some random data.</p> <p>SAMPLE DATA: INPUT: "This is human resource department." OUTPUT: Department Resource Human This Is. INPUT: "To handle yourself use your head and to handle others use your heart." OUTPUT: Yourself Handle Handle Others Heart Head Your Your and Use Use To To.</p> | 04 | |

| | | | |
|---|---|----|--|
| 6 | <p>(Inheritance)</p> <p>Create a class CovidParameters which inherits Patient class. The class CovidParameters has the following members:</p> <ul style="list-style-type: none"> • No argument constructor which initialises CTScore, D-dimer and platelet to zero and also call the parent class constructor. • Parameterise constructor which initialise the class variables as per arguments given and also call the parent class constructor. • float CTScore - which is used to store CT scan score of patients. • float D-dimer - which is used to store D-dimer score of patients. • int platelet - which is used to store platelet count of patient. | 04 | |
| 7 | <p>(Abstract Class)</p> <p>a. Create an abstract class Instrument which is having the abstract function play. Create three more sub classes from Instrument which is Piano, Flute and Guitar. Override the play method inside all three classes printing a message. “Piano is playing tan tan tan tan” for Piano class “Flute is playing toot toot toot toot” for Flute class “Guitar is playing tin tin tin” for Guitar class You must not allow the user to declare an object of Instrument class.</p> <p>Create an array of 10 Instruments. Assign different type of instrument-to-instrument reference. Check for the polymorphic behavior of play method.</p> <p>Create a compartment of a type as decided by a randomly generated integer in the range 1 to 4. Check the polymorphic behaviour of the notice method.</p> | 02 | |
| 8 | <p>(Package and Interface)</p> <p>a. Create an interface Polygon containing the members as given below: void calcArea (); abstract method to calculate area of a particular polygon given its dimensions void calcPeri (); abstract method to calculate perimeter of a particular polygon given its dimensions void display (); method to display the area and perimeter of the given polygon.</p> <p>Create a class Square that implements Polygon and has the following member: float side</p> | 02 | |

| | | | |
|---|--|----|--|
| | <p>Square (float s); constructor to initialise side of square</p> <p>Create another class Rectangle that implements Polygon and has the following member:</p> <p>Rectangle (int len, int bre); constructor to initialise length and breadth of a rectangle</p> <p>Outside the package, create a class that imports the above package and instantiates an object of the square class and an object of the Rectangle class. Call the above methods on each of the classes to calculate the area and perimeter given the side and the length/breadth of the square class and the Rectangle class respectively.</p> | | |
| 9 | <p>(Exception Handling)</p> <p>a) Create a class called MathFunctions. It provides the following functionalities:</p> <p>float getMean() float divideNum() getMean() function takes an integer array as input and provides the mean of the array elements as output. The array element should be in the range of 0 to 100. If the number is less than 0 and greater than 100, raise the custom exception 'InvalidNumber'. Write an appropriate custom exception class for the same. divideNum() function takes two integer numbers as input from the user and provides the result of division (number1/number2). If the number2 entered by the user is zero, raise the built-in exception for 'Divide by Zero'.</p> <p>Create a main class, which will create an object of MathFunctions and call the appropriate method. The user can perform the mean calculation or Division task, allow the user to enter the appropriate choice of operation he/she wants to perform. Show the concept of handling multiple exceptions through a single try block. Also</p> <p>b. Create a class called BankAccount, it has following data members: integer accountNumber, String CustomerName, String AccountType ('Savings' or 'Current'), float balance. Member Functions of the class are : void deposit (float amt); void withdraw (float amt); float getBalance(); deposit(float amt) method allows you to credit an amount into the current balance. If the amount is negative, throw an exception NegativeAmount to block the operation from being performed.</p> <p>withdraw (float amt) method allows you to debit an amount from the current balance. Please ensure a minimum balance of Rs. 1000/- in the account for the savings account and Rs. 5000/- for the current account, else throw an exception</p> | 04 | |

| | | | |
|----|--|----|--|
| | <p>Insufficient Funds and block the withdrawal operation. Also throw an exception Negative Amount to block the operation from being performed if the amt parameter passed to this function is negative.</p> <p>getBalance () method returns the current balance.</p> <p>A constructor to this class will allow you to pass the account number, customer name, account type and opening balance. The minimum opening balance for a savings account is 1000 Rs and for current account, it is 5000 Rs. If the amount entered is less than that, raise the LowBalance exception and prompt the user to enter the opening balance again.</p> <p>Write appropriate custom exception classes.</p> | | |
| 10 | <p>(Multithreading)</p> <p>a) Write a java program that creates two threads (using Thread class). First thread prints the odd numbers till n and thread two prints the even number till n, where n is taken from the user. The output of both the thread should be in format as given below: OddThread: 1 EvenThread :2 The numbers should be printed in sequential order. Use appropriate synchronisation mechanisms if needed.</p> <p>b) Write a stream-based program which will accept Roll Number, Name, Age and Address from user. Age and Roll-no should be numeric. Handle with built-in exception. None of the field should be blank. Handle with custom exception. Ask user, whether to write the data in the file. If answer is yes, then data is saved into a file as an object (User can write many records in the file), otherwise terminate the current program. Write another program to display all the records saved into the file</p> | 02 | |

Nirma University

Institute of Technology

Computer Science and Engineering Department

Practical Policy

B.Tech. Semester - III

Academic Year: 2024-25

| | | |
|-------------------------------|----------|-------------------------------------|
| Course Code & Name | : | 2CS502, Object Oriented Programming |
|-------------------------------|----------|-------------------------------------|

| | | |
|--|---|---|
| Credit Details | : | Lectures-2, Tutorial-0, Practicals-2, Credits-3 |
| Course Co-ordinator | : | Prof Kruti Lavingia |
| Contact No. & Email | : | 079716529572 kruti.lavingia@nirmauni.ac.in |
| Office | : | Mon to Fri: 09:00 am to 4:00 pm, Sat: 9:00 am to 3:00 pm |
| Course Faculty: | | |
| <u>Prof Kruti Lavingia</u> Email: kruti.lavingia@nirmauni.ac.in , Contact No. 07971652572, Office: N-block, Fifth floor NF8. | | |
| <u>Prof. Ajay Patel</u> Email: ajaypatel@nirmauni.ac.in , Contact No. 07971652573, Office: 5th floor N Block NF9 | | |
| <u>Dr. Sanjay Patel</u> Email: sanjay.patel_cse@nirmauni.ac.in , Contact No. 07971652573, Office: 5th floor N Block NF6 | | |
| <u>Dr. Ramesh Ram Naik</u> Email: rameshram.naik@nirmauni.ac.in , Contact No. - 7499064904, Office: 11 th floor E block | | |
| <u>Prof. Anuja Nair</u> Email: anuja.nair@nirmauni.ac.in , Contact No. 07971652563, Office: 5 th floor N Block NF3 | | |
| <u>Dr. Kanu Patel</u> Email: kanu.patel@nirmauni.ac.in Contact No. 9428552448, Office: 11 th floor E block | | |
| <u>Prof.Nirav Patel</u> Email: nirav.patel@nirmauni.ac.in , Contact No. 8511801672, Office: 5 th floor N Block | | |
| <u>Prof.Unnati Mishra</u> Email: 21ftphde63@nirmauni.ac.in , Contact No. 9753727574 Office: 5 th floor N Block | | |

| Rubrics for Regular Evaluation Weightage (0.6) | Rubrics for Final Examination Weightage (0.4) |
|--|---|
| Active involvement in different tasks related to practical / Eagerness to learn | Demonstration of accurate understanding of the objective of practical |
| Regularity in attending practical sessions | Fundamental concepts and technical know-how about practical |
| Preparedness for practical session | Correlation of theoretical concepts with real-life applications |
| Ability to work in a team | Question-answers, writing and presentation skills |
| Originality, completeness, presentability and timely submission of assigned work/laboratory manual | Use of appropriate procedures, tools and techniques to conduct experiments and collect data |

| | |
|-------------------------------------|--|
| Discipline during practical session | |
|-------------------------------------|--|

| Criteria | Satisfactory (0.5 mark) | Adequate (1 mark) | Proficient (1.5 mark) | Outstanding (2 mark) |
|--|---|---|---|--|
| Lab participation (02 mark) | Student arrive late in the lab and rarely participate in lab proceedings. No attempt/desire to learn. | The student has a tendency to arrive late and unprepared. Such unpunctuality or unpreparedness makes it impossible to fully participate. | The student arrives on time to the lab, but may be unprepared and hence could not participate to the full extent. | The student arrives on time with complete preparation and participates in lab proceedings with full enthusiasm. Even eager to explain concepts to fellow batch mates and ready to assist them. |
| Definition of objectives and scope (02 mark) | The student is unaware of the practical objectives and concepts | The student has a difficulty in understanding/explaining key concepts of the practical | The student has a basic knowledge of content but may lack understanding of some of the concepts. | The student demonstrates an accurate understanding of the objectives and concepts. |
| Proper use of procedures (02 mark) | Unable to follow the instructions and performs the experiment. | Follows a limited set of instructions and performs the experiment half-heartedly. | Performs the experiment in a proper manner, however occasionally not following the procedures. | Follows all the instructions given by the instructor and performs the experiment in a perfect manner. Also, influence/emphasise others to follow the procedures. |
| Result analysis and discussion/ Timely Completion of the work (02 mark) | Calculations/Graphs/Quizzes are not complete and not submitted within the given deadline. | Calculations/Graphs/Quizzes are partially complete in a very random/ haphazard or disorganised manner. Work is inaccurate and has a number of errors. | Calculations/Graphs/Quizzes are complete. However, student could do the work more neatly by incorporating all the required information. | Calculations/Graphs/Quizzes are complete and neat. They include all the required details like titles, sketches, units etc. Errors, if any are minimal |
| Question-Answer/Presentation (02 mark) | Unable to answer the questions. Poor language and communication with a number of mistakes. | Answers to the questions are basic and superficial suggesting that concepts are not fully grasped. Language and communication is not clear and fluent, suggesting scope for improvement | Questions are answered fairly well barring a few questions. The language is good. Communication is clear. | All the questions are answered completely and correctly. Language is error-free. Communication is clear and fluent. No grammatical mistakes. |

Rubric for continuous assessment of practical (10 marks)

Note: Copying in any form is not acceptable and will invite strict disciplinary action. Evaluation of corresponding component will be affected proportionately in such cases. Academic integrity is expected from students in all components of course assessment.

Index

| Sr. No. | Title of experiment | Date | Marks (as per rubrics) | | | | | Total Marks (10) | Sign |
|---------|---|------|------------------------|----|----|----|----|------------------|------|
| | | | 02 | 02 | 02 | 02 | 02 | | |
| 1. | Setup Java Programming Environment and Preliminaries and Basic Java Programming | | | | | | | | |
| 2. | Decision-Making Structures and Loops | | | | | | | | |
| 3. | Programs on Operators and Arrays | | | | | | | | |
| 4. | Programs on Classes and Objects | | | | | | | | |
| 5. | String Handling Concepts | | | | | | | | |
| 6. | Inheritance based Programs | | | | | | | | |
| 7. | Abstract class | | | | | | | | |
| 8. | Package and Interface Concepts | | | | | | | | |
| 9. | Exception Handling | | | | | | | | |
| 10. | Multithreading and IO | | | | | | | | |

Lab Faculty Signature and Name