NIRMA UNIVERSITY

Institute:	Institute of International Study
Name of Programme:	Bachelor of Science (Computer Science and Engineering) [2+2 Dual Degree]
Faculty	Faculty of Technology & Engineering
Course Code:	~~1XXXX~~ 2CS502
Course Title:	Object Oriented Programming
Course Type:	Core
Year of Introduction:	2023-24

L	T	Practical Component				C
		LPW	PW	W	S	
2	-	2	-	-	-	3

Course Learning Outcomes (CLO):

At the end of the course, students will be able to –

1. interpret the basic principles of object-oriented programming.
2. develop computer programs to solve real world problems based on object-oriented principles.
3. demonstrate the concept of various exception-handling mechanisms in object oriented programming.
4. develop multi-threaded applications with basic input-output operations and exception handling.

Syllabus: **Total Lecture hours: 30**

Unit	Syllabus	Teaching hours
Unit-I	**Overview of Java**: Features of Java, byte code, Java Development Kit (JDK), Java Virtual Machine (JVM), Introduction to three OOP principles (Inheritance, Polymorphism, Encapsulation), Introduction to classes and methods	02
Unit-II	**Data Types, Variables and Operators in Java:** Arrays: One dimensional array, multi-dimensional array, alternative array declaration statements. Control Statements like Selection statements (i.e if, switch etc.), iteration statements (i.e while, do while, the for-each version of the for Loop, Nested Loops etc.), jump statements (i.e break, continue)	06
Unit-III	**Classes and Methods:** class fundamentals, objects, assigning object reference variables, methods in class, constructors, this keyword, garbage collection, finalize () method, overloading methods, argument passing, access control, static, final, nested and inner classes, command line arguments, variable-length arguments. String Handling: Basics of String handling in Java, String class methods, String Buffer Class methods Inheritances: Basics, member access and inheritance, super class references, using super, multilevel hierarchy, constructor call sequence, method overriding, dynamic method dispatch, abstract classes, Object class Packages and Interfaces: defining a package, finding packages and CLASSPATH, access protection, importing packages, interfaces (defining, implementation, nesting, applying), variables in interfaces, extending interfaces, instance of operator	11
Unit-IV	**Exception Handling**: fundamental, exception types, uncaught exceptions, try, catch, throw, throws, finally, multiple catch clauses, nested try statements, built-in exceptions, custom exceptions (creating your own	5

9.3

exception sub classes).

Unit-V **Multithreading and I/O:** Multithreaded Programming: Java thread model, thread priorities, synchronization, messaging, Thread class, Runnable interfaces, creating a thread(s), Thread class methods, Synchronization, Inter thread Communication, volatile operators. Managing I/O: Streams, Byte Streams and Character Streams, Predefined Streams, reading console Input, Writing Console Output, PrintWriter class, File management classes — 6

Self-Study: The self-study contents will be declared at the commencement of semester. Around 10% of the questions will be asked from self-study contents

Suggested Readings/ 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 I, Volume I—Fundamentals Prentice Hall

Suggested List of Experiments:

No	Title	Hours
1	**(Basic Java Programming)** Hands-on practice on C- Programming: 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 Expected Output: Total number of duplicate elements found in the array is: 2 Duplicate elements:1,2 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 **b)** Write a Java program to display greeting message like: "First Java Program…." on console. **c)** Write a Java program to display all primitive type variables. Also display your name in the last line.	02
2	**(Operators)** **a)** Write a Java Program that check whether user entered number is special number or not. For example,	04

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.

b) Write a Java program that demonstrate the concepts of automatic and explicit type casting.

c) Write a Java program to:

i. check whether a number is odd or even (using if – else statement) ii. check the category of a given character. (Using if…else…if ladder)

ii. check whether a number is prime or not. (Using for loop)

iii. display reverse of a number and check whether it is palindrome or not. (Using while/do while loop)

iv. pattern printing. (Using nested loops)

```
     1
   1 2
  1 2 3
 1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

3 **(Operators and Array)** 04

Write a Java program for the following:

a) Design calculator which contains arithmetic & bitwise operators. Operand(s) and operator must be scan from the user.

b) Given an array of positive and negative numbers. Find if there is a subarray with 0 sum.

Example:
Input: 5
4 2 -3 1 6
Output: Yes
Explanation:
2, -3, 1 is the subarray with sum 0.

4 **(Class and Objects)** 02

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 initializes the object with some default values. Provide public member methods for each of the following:

- 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 the two complex numbers. The method must return complex class object. • display () – It displays the complex number in a+bi format.

The output should be displayed as follows: -
Sum of a1+b1 i & a2+b2 i is: a3+b3 i

9.3

5 **(String Handling)** 04

a) Write a Java Program which asks user to enter a paragraph and perform the following operations:

1. Find total no of sentences in the paragraph and the total number of words in each sentence.
2. 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.
3. Search a word (entered by the user) in the paragraph and print the position of the word (if found) or print appropriate message.

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.

Test your program for the following data and some random data.

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.

6 **(Inheritance)** 04

a) Create a class CovidParameters which inherits Patient class. The class CovidParameters has following members:

- No argument constructor which initializes CTScore, D-dimer and platelet to zero and also call the parent class constructor.
- Parameterize constructor which initialize the class variables as per arguments given and also call the parent class constructor.
- float CTScore - which is used to store CT scan score

30

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.

7 **(Abstract Class)** 02

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.
Create an array of 10 Instruments.
Assign different type of instrument-to-Instrument reference.
Check for the polymorphic behavior of play method.

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.

8 **(Package and Interface)** 02

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.

Create a class Square that implements Polygon and has the following member:
float side
Square(float s); constructor to initialize side of square

Create another class Rectangle that implements Polygon and has the following member:

Rectangle(int len, int bre); constructor to initialize length and breadth of a rectangle

Outside the package, create a class that imports the above package an 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.

9 **(Exception Handling)** 04

a) Create a class called MathFunctions. It provides

following functionalities:
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 build in exception for 'Divide by Zero'.

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

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 withdrad (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.

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 savings account and Rs. 5000/- for current account, else throw an exception InsufficientFunds and block the withdrawal operation. Also throw an exception NegativeAmount to block the operation from being performed if the amt parameter passed to this function is negative.

getBalance() method returns the current balance.

A constructor to this class will allow you to pass account number, customer name, account type and opening balance. 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.

Write appropriate custom exception classes.

10 **(Multithreading)** 02

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 synchronization mechanisms if needed.

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

Suggested Case List: -NA-

cbb
9.3