CSEN1111	OBJECT ORIENTED PROGRAMMING WITH JAVA	L	T	P	S	J	C
		0	0	4	0	0	2
Pre-requisite	Programming with C/Python						
Co-requisite	**None**						
Preferable exposure	**None**						

Course Description:

This course enables the students to gain knowledge on various object-oriented aspects of Java. The course tours the students through classes, inheritance, interfaces, packages, exceptions, generics, graphical programming concepts. The knowledge gained in this course can be applied to develop standalone applications for Android, Real Time Programming etc.

Course Educational Objectives:

- To familiarize object-oriented programming concepts and techniques.
- To illustrate classes and class libraries, developing classes for simple applications.
- To illustrate the usage of Arrays and Strings.
- To demonstrate various types of Inheritance mechanisms.
- To introduce packages applicability and usage of Exceptions.

UNIT 1 **Java Programming Fundamentals** **P - 12 hours**

Java Programming Fundamentals

Java Language, Key Attributes of Object-Oriented Programming, Java Development Kit, Simple Program,Create Blocks of Code, Keywords, Identifiers, The Java Class Libraries.

Data Types and Operators: Java's Primitive Types, Literals, Variables, Scope and Lifetime of Variables,Operators- Arithmetic, Relational, Logical, Bitwise, Assignment. Type conversion in Assignments, Using a Cast, Operator Precedence.

Program Control Structures: if, switch, for, enhanced for, while, do-while, break, continue.

Exercises:

1. Program to read a number from the user and print whether it is positive or negative.

2. Program to solve quadratic equations (use if, else if and else).

3. Take three numbers from the user and print the greatest number.

4. Program that keeps a number from the user and generates an integer between 1 and 7 and displays the name of the weekday.

5. Program that reads in two floating-point numbers and tests whether they are the same up to three decimal places.

6. Program that takes a year from user and print whether that year is a leap year or not.

7. Program to display the first 10 natural numbers.

8. Program to input 5 numbers from keyboard and find their sum and average.

9. Program in Java to display the multiplication table of a given integer.

10. Program in Java to display the pattern like right angle triangle with a number.

Input number of rows : 5

Expected Output :

1

12

123

1234

12345

UNIT 2 Introduction to Classes Objects and Methods P - 12 hours

Introduction to Classes Objects and Methods

Class Fundamentals, Objects creation, Reference Variables and Assignment, Methods, Returning a Value, Using Parameters, Constructors, Parameterized Constructors, new Operator, this Keyword, finalize() method, Wrapper Classes, Parsing, Auto boxing and Unboxing. I/O: Command-Line Arguments, Scanner and Buffered Reader Classes,

A Closer Look into Methods and Classes: Controlling Access to Class Members, passing objects to methods, passing arguments, Returning Objects, Method Overloading, Overloading Constructors, Understanding Static, Variable-Length Arguments.

Exercises:

1. Program to read two numbers and perform the arithmetic operations using methods.

2. Program that performs arithmetic operations with values of type char.

3. Design a class to overload a method compare() to return the greater of two as follows:

void compare(int, int)

void compare(char, char)

void compare(String, String)

4. Program that creates a class Account that stores a variable balance. The class has methods to start account, to deposit money, to withdraw money and tell the current balance amount.

5. Program to implement a Book class that stores the details of a book such as its code, title and price (Use constructors to initialize the objects).

6. Differentiate between static and non-static methods in java.

7. Illustrate the usage of 'this', 'final' and 'finalize' using a java program.

8. Write a java program to implement the concept of dynamic method dispatch

9. How to pass the variable length arguments in java, illustrate with an example program.

10. Write a java program to overload the constructors.

11. Read the command line arguments and print the total number of arguments and itsvalues.

UNIT 3 **Arrays and Strings & Strings** **P - 12 hours**

Arrays and Strings

Arrays: 1D Arrays, Multidimensional Arrays, Irregular Arrays, Array References, Using the Length Member.Arrays class of util package, Array Lists, Vector class

Strings: String class, constructors, length(), string literals, concatenation, toString(), Character extraction,string comparison, searching strings, modifying, data conversion, changing the case, joining, split().

StringBuffer class: constructors, length(), capacity(), ensureCapacity(), setLength(), charAt(), setCharAt(),getChars(), append(), insert(), reverse(), delete(), deleteCharAt(), replace().

Exercises

1. Program for sorting a given list of names in ascending order.

2. Program to multiply two given matrices?

3. Program to find Maximum and minimum value in an array of size "M", passed as argument.

4. Program to read and print an array of size N rows with variable column size .(Hint: Irregular array).

5. Program that copies contents of one array to another using length member.

6. Program to find element from an sorted array using binary search (java.util.package)

7. Program to delete duplicate elements from an array of size 5.

8. Program that reverses an array and stores it in the same array.

9. Program to implement all String methods on a Input String.

10. Convert a given integer array of Size “N” into string.

11. Program to read and print a given string using different methods.

12. Program to reverse the words in a string.

13. Program to read a string and replace all the vowels with a ‘$’ symbol.

14. Program to count the number of occurrences of a search string in a given text string.

UNIT 4	Inheritance and Interfaces	P - 12 hours

Inheritance and Interfaces

Inheritance: Basics, Member Access and Inheritance, Constructors and Inheritance, Using Super, Multilevel Hierarchy, Constructor execution hierarchy, Superclass References and Subclass Objects, Method Overriding, Abstract Classes, Using final.

Interfaces: Fundamentals, Creating and Implementing an Interface, Using Interface References,

Implementing Multiple Interfaces, Extending Interfaces, Nested Interface.

Exercises:

1. Define a class Point with two fields x and y each of type double. Also , define a method

   distance(Point p1, Point p2) to calculate the distance between points p1 and p2 and return the value in double. Use Math.sqrt( ) to calculate the square root.

2. A class Shape is defined with two overloading constructors in it. Another class Test1 is partially defined which inherits the class Shape. The class Test1 should include two overloading constructors as appropriate for some object instantiation. You should define the constructors using the super class constructors. Also, override the method calculate() in Test1 to calculate the volume of a Shape.

3. Create a class named 'Member' having the following members: Name, Age, Phone number, Address, Salary. It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.

4. Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This I rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' clas by the object of 'Square' class.

5. Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call

   - method of parent class by object of parent class
   - method of child class by object of child class
   - method of parent class by object of child class

6. Create a class telephone with ( ) , lift ( ) and disconnected ( ) methods as abstract methods create another class smart telephone and demonstrate polymorphism

7. Design a vehicle class hierarchy in Java, and develop a program to demonstrate Polymorphism.

8. Write a program to find the roots of a quadratic equation using interface and packages.

   - Declare an interface in package Quad1
   - Declare another package Quad2 and implement the interface

9. Write a Program to generate Fibonacci Series by using Constructor to initialize the Data Members.

10. Develop a program to demonstrate multiple inheritance through interface.

UNIT 5	**Packages and Exception Handling**	**P - 12 hours**

Packages: Package Fundamentals, Member Access, Importing Packages, Static import.

Exception Handling: Exception Hierarchy, Fundamentals, Consequences of an Uncaught Exception, Handling errors, Multiple Catch, Throwing and Rethrowing an Exception, Throwable, using finally, using throws, Creating Exception Subclasses.

Exercises:

1. Program to demonstrate the visibility of members in subclasses of same and different packages.
2. Program to create a user defined package in Java.
3. Program to find the roots of a quadratic equation using interface and packages.
   - Declare an interface in package Quad1
   - Declare another package Quad2 and implement the interface
4. Define a Interface Polygon in package pack1. create a class triangle from Polygon in package pack2, override method to calculate area of the triangle and raise an exception if it is an equilateral triangle.

Note : Exception has to be defined in package pack3.

5. Develop a program to demonstrate exception handling by using THROW, MULTIPLE CATCH & FINALLY statements.
6. Create a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user-defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException". Define the two exception classes.
7. Program to throw a user defined exception for employee details
   - If an employee name is a number, a name exception must be thrown.
   - If an employee age is greater than 50, an age exception must be thrown
8. Program to demonstrate nested exception.
9. Create an Account class with data members accno, name,bal. Include methods deposit(), withdraw(). Raise an exception when balance in account is less than 1000.
10. Create a Student class with data members Rollno, Name, marks in subjects. Include methods to compute average. Raise an exception if the student has more than 2 backlogs.

TextBooks:

1. Herbert Schildt, Dale Skrien, Java Fundamentals A Comprehensive Introduction, 1/e, Tata McGraw Hill, 2017.
2. Herbert Schildt, The Java complete References, 9/e, Tata McGraw Hill,2014.

References:

1. Y.DanielLiang , An Introduction to JAVA Programming, 10/e, Tata McGraw Hill.
2. Kathy Sierra, Head First Java, 2/e, Shroff Publishers, 2012.
3. Balagurusamy , Programming with JAVA, 2/e, Tata McGraw Hill, 2014.

Course Outcomes:

After successful completion of the course the student will be able to:

1. describe the data types, operators and control structures
2. understand the concepts of Object Oriented Programming
3. make use of Arrays and Strings related operations
4. apply features of OOPs to build real time applications
5. demonstrate the ease of handling various scenarios of program execution without abrupt interruption

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1		1													
CO2	1	2												1	
CO3	1	2	1	1									1	2	1
CO4	1	2	2	1									1	2	
CO5	1	2	1	1									1	2	1

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

APPROVED IN:

BOS : 06-09-2021 **ACADEMIC COUNCIL: 01-04-2022**

SDG No. & Statement:

SDG Justification: