Asansol Engineering College

Sub. Name **Object Oriented Programming** Sub. Code: **PCC-CS593** Dept- **IT**

CHAPTER	TITLE
A: Basic Java programs using Encapsulation concept	1. Write a program to check whether a number is ODD or EVEN. ***Test Data:*** **Expected Output:** 32 is an EVEN number **Expected Output:** 133 is an ODD number 2. Generate all the ODD numbers from 1 to 100. 3. Write a program to check whether a number is Prime or Not. ***Test Data :*** **Expected Output:** 32 is a Not a Prime number **Expected Output:** 17 is a Prime number 4. Generate all the Prime numbers from 1 to 100. 5. Write a program to perform the following operation between two operands of integer type, the operation includes Addition, Subtraction, Division and Multiplication. ***Test Data :*** **Expected Output:** 20 + 3 = 23 20-3 = 17 20 * 3 = 60 20 /3 =6 **Expected Output:** 20 + 0 = 20 20-0 = 20 20 * 0 = 0 20 /0 = Undefined 6. Define a class Calculator which has two integer members. Define add(), sub(), multi(), divi() , show() methods to perform Addition, Subtraction, Multiplication , Division and Display operations. [Don't pass the parameter in the methods]. Create an object and call the methods in the correct order. 7. Define a class Calculator2 which has two integer members. Define add(int,int), sub(int,int), multi(int,int), divi(int,int) , show() methods to perform Addition, Subtraction, Multiplication , Division and Display operations. Create an object and call the methods in the correct order.
B: Message Passing Concept	1. Define a class VOperations. Define initializer(), linearSearch(), binSearch(), bubbleSort(), show() methods to initialize the array, **Arr**, with {11,6,77,8,5,44,6,9,442,86,73,49,68,82}, to search an element **se** using Linear Search, to search an element se using Binary Search, to perform the Bubble Sort on the array, to display the elements in the array, Arr, respectively. 2. Define a class Matrix that contains a method, sumDiagonals(). This method will find out the sum of left and right diagonals of a matrix of size M x N. 3. Define a class Sparse. Take a sparse matrix of Size M x N and represent that sparse matrix in Triplet format.

C: Concept of various scopes of members & object reference

1. Write a program to show how ***'this'*** keyword is used to differentiate between an instance variable and a local variable with the same name.
2. Write a program to show the order of execution between local block(s), static block(s), instance block(s), and constructor(s).
3. Write the definition for a class called **Complex** that has floating point data members for storing real and imaginary parts. The class has the following member functions:
   void set(float, float) to set the specified value in the object
   void disp() to display complex number object
   complex sum(complex) to sum two complex numbers & return complex number

   a. Write the definitions for each of the above member functions.
   b. Write the main function to create three complex number objects. Set the value in two objects and call sum() to calculate the sum and assign it to the third object. Display all complex numbers.
4. Write a program in JAVA to show that the object is passed by reference but primitive data types are passed by value.
5. Write the definition for a class called Rectangle that has floating point data members: length and width. The class has the following member functions:
   void setlength(float) to set the length of data member
   void setwidth(float) to set the width data member
   float perimeter() to calculate and return the perimeter of the rectangle
   float area() to calculate and return the area of the rectangle
   void show() to display the length and width of the rectangle
   boolean isSameArea(Rectangle) that has one parameter of type Rectangle. This method returns True if the two Rectangles have the same area, and returns False if they don't.

   a. Write the definitions for each of the above member functions.
   b. Write the main method to create two rectangle objects. Set the length and width of the first rectangle to 5 and 2.5. Set the length and width of the second rectangle to 5 and 18.9. Display each rectangle and its area and perimeter.
   c. Check whether the two Rectangles have the same area and print a message indicating the result. Set the length and width of the first rectangle to 15 and 6.3.

   Display each Rectangle and its area and perimeter again. Again, check whether the two Rectangles have the same area and print a message indicating the result.

6. Write the definition for a class called **DOB** that has integer primitive data members: day, month, year. The class has the following member functions:
   void setDOB(int, int, int) to set the Date of Birth
   void show () to display the Date of birth in DD/MM/YYYY format
   DOB findAge(DOB) has one parameter of type DOB. This method will find the age difference and will return DOB object. Print all DOB objects' data.

D: Concept of Compile time polymorphism (Method Overloading)

1. Design a class **RailwayTicket** with the following description:

   Data Members Purpose

   String name : To store the name of the customer

   String coach : To store the type of coach the customer wants to travel

   long mobno : To store the customer's mobile number

   int amt : To store the basic amount of ticket

   int totalamt : To store the amount to be paid after updating the original amount

   Member Methods *Purpose*

   void accept() : To take input for name, coach, mobile number, and amount

   void update() : To update the amount as per the coach selected (extra amount to be added in the amount as per the table below)

   void display() : To display all details of a customer such as a name, coach, total amount, and mobile number

Type of Coaches	*Amount*
First_AC	700
Second_AC	500
Third_AC	250
Sleeper	None

   Write a main method to create an object of the class and call the above member methods.

2. Define a class called **ParkingLot** with the following description:

   Data Members Purpose

   int vno To store the vehicle number

   int hours To store the number of hours the vehicle is parked in the parking lot

   double bill To store the bill amount

   Member Methods Purpose

   void input( ) To input the vno and hours

   void calculate( ) To compute the parking charge at the rate of Rs3 for the first hour or the part thereof and Rs1.50 for each additional hour or part thereof.

   void display() To display the detail

   Write a main method to create an object of the class and call the above methods.

3. Design a class to overload a function volume () as follows:

   double volume(double r) — with radius (r) as an argument, returns the volume of the sphere using the formula:

   V = (4/3) * (22/7) * r * r * r

   double volume(double h, double r) — with height(h) and radius(r) as the arguments, returns the volume of a cylinder using the formula:

   V = (22/7) * r * r * h

   double volume(double 1, double b, double h) — with length(l), breadth(b), and height(h) as the arguments, returns the volume of a cuboid using the formula:

   V = l*b*h or (length * breadth * height)

4. Design a class to overload a function series () as follows:

   double series(double n) with one double argument and returns the sum of the series. sum = (1/1) + (1/2) + (1/3) + ... + (1/n)
   double series(double a, double n) with two double arguments and returns the sum of the series. sum = $(1/a^2) + (4/a^5) + (7/a^8) + (10/a^{11})$ + ... to n terms

5. Write a class with the name **Perimeter** using function overloading that computes the perimeter of a square, a rectangle, and a circle.

   The perimeter of a square = 4 * s
   The perimeter of a rectangle = 2 * (l + b)
   The perimeter of a circle = 2 * (22/7) * r

6. Write a class with the name Area using function overloading that computes the area of a parallelogram, a rhombus, and a trapezium.

   Area of a parallelogram (pg) = base * ht

   Area of a rhombus (rh) = (1/2) * d1 * d2
   (where d1 and d2 are the diagonals)

   Area of a trapezium (tr) = (1/2) * (a + b) * h
   (where a and b are the parallel sides, h is the perpendicular distance between the parallel sides)

E: Concept of Compile time polymorphism (Constructor Overloading)

1. Write a program to print the names of students by creating a **Student** class. If no name is passed while creating an object of the Student class, then the name should be "Unknown", otherwise the name should be equal to the String value passed while creating the object of the Student class.
2. Create a class named **'Rectangle'** with two data members- length and breadth and a method to calculate the area which is 'length*breadth'. The class has three constructors which are:
   1 - having no parameter - values of both length and breadth are assigned zero.
   2 - having two numbers as parameters - the two numbers are assigned as length and breadth respectively.
   3 - having one number as a parameter - both length and breadth are assigned that number.

   Now, create objects of the 'Rectangle' class having none, one, and two parameters and print their areas.

3. Create class **MyNumber** with only one private instance variable as a double type. Include the following methods (include respective **constructors**) isZero( ), isPositive(), isNegative( ), isOdd( ), isEven( ),isPrime(), isAmstrong() the above methods return boolean primitive type. getFactorial(), getSqrt(), getSqr(), sumDigits(), getReverse() the above methods return double primitive type. void listFactor(), void dispBinary() will show all the factors of the number and the binary equivalent of the number.
4. Implement a **Student** class with the following fields, constructors and methods:
   Fields:
   name;
   totalScore;
   numberOfQuizzes;
   Constructors:
   public Student(String name, double score)
   public Student(double score, String name)
   public Student(String name)
   Methods:
   public String getName()
   public double getAverage() //this should return zero if no quiz has been taken.
   public double getTotalScore()
   public void addQuiz(double score)
   public void printStudent() //this should print the student's name and average score
   Write a java program that reads a student name and use the **Student** class to create a **Student** object. Then read the scores of the student in three quizzes and **add** each to the **totalScore** of the student using **addQuiz**() method and print the student object.

5. Write a program to show Constructor Chaining in the same class.

F: Concept of Code Reusability (Inheritance)

1. Create a class **Parent** with a method that prints "This is parent class" and its subclass **Child** with another method that prints "This is child class". Now, create an object for each of the classes and call
   1 - method of parent class by the object of the **Parent** class
   2 - method of child class by the object of **Child** class
   3 - method of parent class by the object of **Child** class
2. In the above example, declare the method of the **Parent** class as private and then repeat the first two operations (You will get an error in the third). It shows that the private member is only accessed within the same class and the private member is not inherited.
3. Create a class named **'Member'** having the following members:
   Data members
   1 - Name
   2 - Age
   3 - Phone number
   4 - Address
   5 – 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 the name, age, phone number, address, and salary to an employee and a manager by making an object of both of these classes and printing the same.
4. Design a class **Person** with the following specification
   - zero-argument constructor
   - nationality()
   - birthPlace()

   The **Employee** class inherits from the **Person** with the following specification
   - zero-argument constructor
   - organisationName()
   - workPlace()

   The **Manager** class inherits from **Employee** with the following specification
   - zero-argument constructor
   - noOfSubordinates()
   - managingPlace()

   Create the necessary object to show the concept of ***multi-level*** inheritance.
5. Design a class **Person** the following specification
   - zero-argument constructor
   - One argument constructor
   - two argument constructor

   The **Employee** class inherits from the Person with the following specification
   - zero-argument constructor
   - One argument constructor, calls its parent's single argument constructor
   - two-argument constructor, it calls its parent's double argument constructor

   The **Manager** class inherits from Employee with the following specification
   - zero-argument constructor
   - One argument constructor, calls its parent's single argument constructor

- two-argument constructor, it calls its parent's double argument constructor

Create the Three objects of the **Manager** class with Zero, Single & Double argument constructors and conclude the order of execution of different constructors.

G: Concept of Dynamic Polymorphism

1. Design a class **Animal** with the following specification
   - sound(), it prints "Sound of Animals, which varies"
   - hasLife(), it returns True
   - hasTail(), it returns True
   - noOfEyes(), it return 2
   - noOfLegs(), it return 4

   Design another class **Dog** which is inherited from **Animal** class with the following specification
   - sound(), it prints "Barking"

   Design another class **Tiger** which is inherited from the **Animal** class with the following specification
   - sound(), it prints "Roaring"

   Create the Object of Dog & Tiger and the necessary methods to illustrate the concept of Method overriding.

2. Follow the following code and write a program to show that 'super' keyword is used to access the super class **data member** (maxSpeed).

```
/* Base class vehicle */
class Vehicle
{
    int maxSpeed = 100;
}
/* sub class Car extending vehicle */
class Car extends Vehicle
{
    int maxSpeed = 140;
  void display(){ ……..}

}
```

```
Expected Output:
My parent's speed was : 100 KM/H
My Speed is           : 140 KM/H
```

3. Follow the following code and write a program to show that 'super' keyword is used to access the super class **method member** (**message()**).

```
/* Base class Person */
class Person
{
    void message()
    {
        System.out.println("This is person class");
    }
}
/* Subclass Student */
class Student extends Person
{
    void message()
    {
```

```
            System.out.println("This is student class");
        }
       // Note that display() is only in Student class
        void display()
        {
            ……….
        }
}
```

Expected Output:

```
My parent's Message was : This is person class
My Message Is           : This is student class
```

4. Design a class **Animal** with the following specification
   - sound(), it prints "Sound of Animals, which varies"

   class **Dog** is inherited from Animal class with the following specification
   - sound(), it prints "Barking"

   class **Tiger** is inherited from Animal class with the following specification
   - sound(), it prints "Roaring"

   With the help of these classes show the concept of Dynamic Method Dispatching.[If the user gives D then it will print "Barking" , T then it will print "Roaring" else Exit from the program]

H: Concept of Abstraction using Interface and abstract class

1. Design an abstract class **Bank** that has the following members:
   abstract double **getRateOfInterest**() . // would return Rate of interest
   String **getBankName**() // prints "No Specilised Bank"

   MySBI & **MyPNB** are two classes that are inherited from the Bank class. **MySBI** provides an 8.35 interest rate. **MyPNB** provides a 7.35 interest rate. These two inherited classes also override **getBankName**() method and it prints their names. Write a program to show how the abstract class's reference is used to refer MySBI & MyPNB class's reference and also call the required methods.

2. Design an interface, **Calculator**

```
interface Calculator{
   public void sum();public void sub();
   public void multi();public void div();
}
```

   Implement this interface in a class named as **MyCalculator** and show how the interface's reference is used to refer to a sub-class's reference. Call all the necessary methods.

3. Follow the relationship between interfaces and class. Complete the code for the Demo class to establish the fact that the interface can be extended by another interface.

```
interface Inf1{
   public void method1();
}
interface Inf2 extends Inf1 {
   public void method2();
}
public class Demo implements Inf2{
```

```
…..
…..
}
```

4. Write a program to show that Java allows multiple inheritances between interfaces. ***Hints:***

```
public interface X {
    public void methodX();
}

public interface Y {
    public void methodY();
}

public interface Z extends X, Y {
    public void methodZ();
}
```

```
public class Sub implements X, Y, Z
{
    public void methodX() { }

    public void methodY() { }

    public void methodZ() { }
}
```

5. Write a program to show how and which of the four data members (with different Access Specifiers) of a particular class are accessed in the -
   i. Subclass of the same package.
   ii. Non-subclass of the same package.
   iii. Subclass of a different package.
   iv. Non-subclass of a different package.

I: Collection Framework (ArrayList)

1. Perform the following operations on the Array List named as first.
   - Insert the following members [10, 20,30, 40,50]
   - Show all the elements in the single line format: [10, 20,30, 40,50]
   - Add 60 at the 4th index position
   - Show all the elements in separate lines
   - Check whether 30 and 55 are present or not; if present then what is the index position; if not present then what will be the output
   - Create another ArrayList named as second. Add [-60,-70,-80] into the second ArrayList using the following command
     …….. second = new ArrayList<> (Arrays.asList(-60,-70,-80));
   - Add second arraylist to the first arrylist
   - Show all the elements present in the first arraylist in a single line format
   - **Insert** 999 at index position 3 of first
   - Show all the elements present in the first arraylist in a single line format
   - Do the addition of 0th and 2nd index positioned element of the first ararylist
   - **Modify** the 0th element of the first as 111
   - Show all the elements present in the first arraylist in a single line format
   - **remove** element at index 4 of first
   - Show all the elements present in the first arraylist in a single line format
   - **remove** element, -80, from the first arraylist
   - **Count** how many elements are present in the first
   - **Sort** the first arraylist in ascending order and descending order. Show all the elements present in the first arraylist in a single line format after each sorting.

2. Design a Student class with name ,roll, age and marks fields. Write constructor to initialize all the members. All getter and setter methods should be included. Create an ArrayList called as db. Add the following data to db

Name	Roll	Age	Marks
Riya	56	23	89
Jackson	24	28	78
Amina	48	23	80
Soumya	35	21	75

Override `public String toString() { }` method in such a way that this method will return a string containing the following information in the following way. E.g.

Riya - - 56 - - 23 - - 89

Print all the data in different lines. Sort the data and print the data based on

*Name * Roll * Age *Marks

J: Collection Framework (LinkedList)

1. Implement a Customer Queue using LinkedList. Class Customer will have customer id (cid) , name (cname), age(cage). Design necessary constructor(s). You can use the necessary getter setters. Your program should have menu-driven options like 1 for Enqueue (insert an element into the queue), 2 for Dequeue (deleting element from the queue) 3 for Display (print all the customers' data), 4 for modifying customer data
   - Insert the following data

Id	name	age
3	AAA	45
1	BBB	34
7	KKK	21
5	FFF	30
4	CCC	40

   - Print the data in id -- name -- age format
   - Delete one customer from the queue and print the data of the deleted customer. Also, show all the data on the queue
   - Modify the customer's data

	Old	New
Id	7	7
Name	KKK	SSS
Age	21	25

   - Print all the data (each customer's data in each line)

   Using LinkedList implement a STACK of String. Perform push, pop, and display operations.

K: Collection Framework (PriorityQueue)	1. Using PriorityQueue implement a min priority queue and max priority queue of Integer. 2. Using PriorityQueue implement a min priority queue of Book. Book is a class that contains bookname, bookid, bookprice fields. Book **will** implement a Comparable interface. Your min priority queue will be based on bookprice. Add remove display the elements of the priority queue. 3. Using PriorityQueue implement a min priority queue of Book. Book is a class that contains bookname, bookid, bookprice fields. Book **will not** implement a Comparable interface. Your min priority queue will be based on bookprice. Add remove display the elements of the priority queue.
L: Collection Framework (HashSet)	1. Use HashSet to perform the following operations: a. create a HashSet object with the data: 88, 44,2,90,75,12,52,75 b. print the data c. segregate the odd and even data and store them in ODD & EVEN HashSet respectively. d. print the data of ODD & EVEN 2. Use TreeSet to perform the following operations: a. create TreeSet object with the following data: 10,2,5,8,100,20,45 b. print all the data c. Print the equal or closest greatest element of the 5 d. Print the closest greatest element of the 5 e. Print the equal or closest least element of the 5 f. Print the closest least element of the 5 g. Print the group of elements that are less than the 34 h. Print a set of elements that are greater than or equal to the 34 i. print the highest number of the tree j. print the lowest number of the tree k. delete the first element of the tree l. delete the last element of the tree print how many elements are present in the tree

M: Collection Framework (HashMap)	1. Use HashMap to store the following information: (userId, UserName)➔ (1,"arup"),(2,"bimal"),(3,"Chandan"),(4,"dola"),(5,"esha"),(6,"rakesh") and do the following operations a. Find the userId of dola b. Find the UserName whose Id is 5, give a suitable message c. Find the UserName whose Id is 56, give a suitable message d. Print all the data using EntrySet e. remove rakesh f. Change the Name of userId 4 as Manisha 2. Using HashMap print the number as words. e.g., 1234 will be printed as One Thousand Two Hundred Thirty Four
N: Concept of Exception Handling	1. Write a program to show the uses of the 'try-catch' block. 2. Write a program to show the uses of the nested 'try-catch' block. 3. Write a program to show the uses of 'throw' & 'throws'. 4. Write a program to show the uses of the 'finally' block. 5. Define a user define an exception & use it.
O: String Handling in Java	1. Write a program that will accept a string as input and count the number of words, digits, vowels & consonants in the string. 2. Create a class named **'Student'** having the following members: Data members: 1 – Name: String 2 – Age: integer 3 - Phone number: String 4 – Address: String 5 – Marks: integer Create an array of 10 Students and perform the following operations: a. Searching for a Name b. Print the name of the student who got Maximum marks. c. Print the detailed information of a particular student. d. Print the detailed information of all the students Design the necessary instance methods and by writing a Menu driven program call all the necessary methods to perform the required operation. 3. Write a program to check whether a string is palindrome or not 4. Take one name as input and print its initials with the full title. (e.g., Sachin Ramesh Tendulkar will be as S.R. Tendulkar)

P: Design Patterns	1. Use Factory method Design Pattern to create Mobile Gadgets (Mobile, Smart Watch, Smart Goggles) 2. Using MVC architecture design the program for the following problem. Suppose you have to maintain an Electricity Bill for 100 customers. Customers will have Name, Id, Unit, and Bill Amount fields. You can add more fields if needed. You have to generate an Electric bill for a customer. Design a menu-driven program to implement this problem.
Q: Multi Threading	1. Write a program to implement the concept of threading by extending **Thread** Class 2. Using **Thread** class create multiple threads. 3. Using **Runnable** interface class create multiple threads which will be operated on same object. 4. Write a program to that the higher priority thread gets more processor than lower priority thread. 5. Using thread create race condition situation & overcome from that situation
R: GUI programming	1. Write a program to show the life cycle of Applet. 2. Write a JApplet program to display the "Hello World "in the browser. 3. Write a program using an applet that will add two buttons named Green & Blue. Whenever the Green button will be clicked the background will be changed to green & when the Blue button will be clicked the background will be changed to blue. 4. Write a program using an applet that will accept two integers and perform the following operation: Addition, Subtraction, Multiplication, & Division. [You have to create four buttons named as ADD, SUB, MULTI, & DIV. Whenever ADD button is clicked then addition will be performed on the given numbers & the result will show.] 5. Write a program using applet that will accept one string. You have to check whether it is palindrome or not.