



COMP 20043

OOPM

1. Write a Java program that contains a class named Circle with an instance variable: radius. The class should have the following methods: 1. assignRadius(): This method assigns a value to the radius of the circle using parameter passed from main method. 2. area(): This method calculates and returns the area of the circle(Use the formula $3.14 * \text{radius} * \text{radius}$). 3. circumference(): This method calculates and returns the circumference of the circle(Use the formula $2 * 3.14 * \text{radius}$). Additionally, create another class named CircleOperations that contains the main method. Within the main method: Create an object of the Circle class. Use the object to call the assignRadius method and assign a sample value to the radius. Calculate and display the area of the circle using the area method. Calculate and display the circumference of the circle using the circumference method.
2. Write a Java program that contains a class named Student with instance variables for name, mark1, mark2, mark 3, mark4.The class should have four methods 1. Setvalues() for setting the values for class variables using parameters,2. Calculatetotalmarks() to calculate and return total marks, 3.calculategrade() to calculate and return student grade (e.g., A, B, C or fail based on totalmarks, if totalmarks > 90, grade is A , if totalmarks between 90 and 70 , grade is B, if totalmarks between 50 and 70, grade is c , else F) and 4. Display() to print these values respectively. Additionally, create another class named StudentOperations that contains the main method. Within the main method: Create objects of the Student class for multiple students. Set their names, and marks. Calculate totalmarks and display the grade of each student
3. Write a Java program that contains a class named Employee with two instance variables: Salary and Bonus. The class should have the three methods: inputvalues(double s,double a). This assigns values to the variable Salary and Allowance using the values passed from the main method. calculateTotalSalary(). This method calculates and returns the total salary. The formula: totalsalary = Salary + Bonus. Display(). This method displays the salary, allowance and totalsalary. Create another class named EmployeeMain that contains the main method. Within the main method: Create an object of the Employee class. Using the object, call all methods, inputvalues(), calculateTotalSalary() and display().
4. Define a class called BankAccount with two data members: balance and Account Number .This class should also have the following. a) constructor to initialize the variables with values passed from main method using parameters. b) set()and get() methods for setting value and returning

value for the class variables. c) method deposit() to add an amount to the balance. d) a method withdraw() to deduct an amount from the balance. Define another class that hosts the main method. Inside the main method: Create an instance/object of the BankAccount class. Call all the methods using the instance/object created to display their functionality.

5. Define a class called Employee with four data members: name, id, position, and salary, and two member functions: void input() to assign input values using parameters passed from the main method and void display() to display the values. Appropriate data types are to be chosen for the data members. Define another class that hosts the main method. Inside the main method: Create an instance/object of the Employee class. Call the input(), display() methods using the instance/object created.
6. Define a class called Product with three data members: productId, name, and price. Appropriate data types are to be chosen. The class should also have the necessary **set** and **get** member functions to set the values for the data members and to return the values of the data members respectively. Define another class that hosts the main method. Inside the main method: Create two instances/objects of the Product class. Call the **set** and **get** methods using the instance/objects created to display their functionality.
7. Create a class called Vehicle with variables brand and model, as well as methods to set and get the values of these variables. Then, create a subclass called Car that has an additional variable numDoors and methods to set and get its value. Define another class that hosts the main method. Inside the main method: Create two instances/objects of the Car class. Call the **set** and **get** methods using the instance/objects created to display their functionality.
8. Write the code for each class and include the necessary methods as described below:
Animal class: Method: void eat() - prints "Animal is eating." ,Method: void sleep() - prints "Animal is sleeping."
Mammal class (inherits from Animal): Method: void giveBirth() - prints "Mammal is giving birth."
Bird class (inherits from Animal): Method: void fly() - prints "Bird is flying."
Reptile class (inherits from Animal): Method: void crawl() - prints "Reptile is crawling."
Dog class (inherits from Mammal): Method: void bark() - prints "Dog is barking."
Eagle class (inherits from Bird): Method: void hunt() - prints "Eagle is hunting."
Snake class (inherits from Reptile): Method: void poison() - prints "Snake is poisonous."
Additionally, create a main method in a class named InheritanceHierarchy to demonstrate the behavior of the different animal types.
9. Create a Parent class named *Employee*. The class should have three instance variables: empname, empid and salary. Also Two methods: getData() and setData() . setData() function should assign some value for all the variables using parameters. getData() function should display/print the values of all the variables. Create a derived class named Manager of Employee class. The class should have One variable: Bonus. Also two methods : getBonus() and setBonus() . setBonus function should set the value for Bonus. getBonus() method should display the values of Bonus.

Create a main class in which you need to create an 2 objects of class manager . Call all the functions of both parent and child classes to check their functionalities.

10. Design a Java program using the concept of method overloading in a class named Shape. The class should have: an attribute to store volume.

The class also should have three overloaded volume() methods and one display method
volume(double length, double breadth, double height) : To calculate volume for cuboid using formula, $\text{volume} = (\text{length} * \text{breadth} * \text{height})$,

volume(double side) : To calculate volume for cube using formula, $\text{volume} = (\text{side} * \text{side} * \text{side})$

volume(double radius, double height) : To calculate volume for cylinder using formula, $\text{volume} = 3.14 * \text{radius} * \text{radius} * \text{height}$

display() to print the calculated volume.

In a separate class ShapeMain, create an instance of the Shape class, calculate volume for various shapes using the overloaded volume() methods, and display the results.

11. Create a class called "Employee" with attributes empName, empID, basic salary and House allowance. Implement the **concept of constructor overloading** by having a parameterized constructor and non- parameterized constructor to initialize the class attributes. The class should also have a method called calculatetotalsalary() which calculates and returns the totalsalary(total of basic salary and house allowance).Display() method must print the value of the all class variables and total salary.

In a separate class, create two instances/objects of the Employee class, one using parameterized constructor and other using non parameterized constructor. Display the information of the two employees by calling the display method.

12. Design a Java program using the **concept of method overloading** in a class named Calculator.

The class should have an attributes to store final calculation results. The class should have 4 multiply methods that can multiply two integer numbers, three integer numbers, 2 double numbers and 3 double numbers respectively. A method to display the calculated result should also be available . In a separate class, create an instance of the Calculator class, perform various multiplication operations using the overloaded Multiply methods, and display the results.

13. Create an **abstract class called Shape** that contains one data member color, two abstract methods called calculateArea() and displayinfo(), which is implemented by Circle class and Rectangle class that extends from the Shape class. Circle class should have two variables radius, and area, one parameterized constructor to give values for the variables color and radius. It should also implement calculateArea() method using formula $\text{area} = 3.14 * \text{radius} * \text{radius}$. It should implement displayinfo() to display , the color, radius and area. Rectangle class should have three variables width and height and area, one parameterized constructor to give values for the variables color , width and height. It should also implement calculateArea() method using formula $\text{area} = \text{width} * \text{height}$. It should also implement displayinfo() to display all its data. Create a main class, create one object each for Circle and Rectangle classes and call the methods respectively

14. Create an abstract class called BankAccount that provides a common structure for all types of bank accounts. It should have the data members, account id and account type, balance amount. It should have two methods to initialize the values of data members and display the

value of data members. It should also include an abstract method calculateInterest() to calculate and display the interest, which is specific to each account type.

Create subclasses SavingsAccount and CurrentAccount that extends BankAccount. They should provide implementations for the calculateInterest() method, tailored to the specific account types using the formula given below

SavingsAccountInterest = 2% of balance amount , CurrentAccountInterest = 0% of balance amount.

In the Main class, create Objects of SavingsAccount and CurrentAccount, and call the get and set methods and calculateInterest() method to calculate the interest for each account type.

15. Create an abstract class called “ InputOutput class” that has two abstract methods. Void getData() and void putdata(). Create a class Teacher that extends from InputOutputClass with attributes teacher id, teacher name, salary. Implement the methods getData() and putdata() to assign values to the Teacher class variables and display the values respectively. Create another class Student that extends from InputOutputClass with attributes Student id, Student name, GPA. Implement the methods getData() and putdata() to assign values to the Student class variables and display the values respectively. Create a main class , create one object each for Teacher and Employee classes and call the methods respectively
16. Write a Java program to implement the **concept of polymorphism** using method overloading in class called **Box**. The class should have 3 instance variables called **length** and **breadth** and **height**. Also **three setValues()** methods to set initial values to the variables. The first set method has **no parameters** and should initialize the instance variables with some values. The second set method should have **three parameters** to initialize the variables using the values passed from main method. The third method should have **one parameter** which should be used to set the values for all the three class variables. The class should also have a method called as **display()** which displays the length, breadth and height of the box and also a message whether it is a normal box or a cube. Create another class that hosts the main method. Create an object of the Box class which invokes the all the three **setValues()** methods. Call the **display()** method also.
17. Create an **interface** called **Tax** with a data member Taxrate that has value of .05. It has three member functions
 1. calcTax () to return calculated tax (tax = salary* taxrate)
 2. CalcNetsalary () to calculate and return net salary (netsalary = gross salary- tax),
 3. Display() to display gross salary, Tax and net salary.Create a class Salary that implements interface tax. This class has one data member for gross salary. One constructor to assign value to the data member using parameter. Create a main class which has main method. Create an object of Salary class. Call the methods display to display gross salary, Tax and net salary .
18. Create an **interface** called **Bill** . It has three member functions
 1. calcTax () to return calculated tax (tax = Price* taxrate)
 2. calcDiscount() return calculated discount (discount = Price* discontrate)
 3. calcTotalPrice () to calculate and return Total Price(TotalPrice= Price +tax-discount),
 - 4 Display() to display Price, Tax, Discount and TotalPrice.

Create two classes IndustryProduct and HouseHoldProduct that implements interface Bill. This class has two data members for Price and DiscountRate . One method assignvalue to assign value to the data members using parameter. Implement the 4 methods of the interface in these two classes.

Create a main class which has main method. Create an object of IndustryProduct and HouseHoldProdcut class. Call the methods display to display Price, Tax, Discount and TotalPrice.