

Q1

Question:

Create two Java classes named "Employee" and "ParttimeEmployee". The Employee class should have two attributes: name (String) and salary (double). Create a constructor that accepts values for both attributes and a method named "displayInfo()" which displays the employee's name and salary. **ParttimeEmployee** class should be extended from the Employee class and should have a specific attribute (int) OTHrs. Create an overloaded constructor which supports the following object creation. Implement a method called "displayEmpInfo" to display all the details of a parttime employee including name, salary, and OT hours.

```
ParttimeEmployee emp1 = new ParttimeEmployee ("Alice", 20000.00, 7);
```

Create a class called "EmpApp" with the main method. Create 2 **ParttimeEmployee** objects and display the details by calling the **displayEmpInfo** method.

EmpApp class

```
public class EmpApp {  
    public static void main (String[] args) {  
  
        ParttimeEmployee emp1 = new ParttimeEmployee ("Alice", 20000.00, 7);  
        ParttimeEmployee emp2 = new ParttimeEmployee ("John", 34000.00, 2);  
  
        emp1.displayEmpInfo();  
        System.out.println();  
        emp2.displayEmpInfo();  
  
    }  
}
```

Employee class

```
public class Employee {  
    String name;  
    double salary;  
  
    public Employee (String newname, double newsalary ) {  
        name=newname;  
        salary=newsalary;  
    }  
  
    void displayInfo() {  
        System.out.println("Employee name: " +name );  
        System.out.println("Employee Salary: " +salary );  
    }  
}
```

ParttimeEmployee class

```
public class ParttimeEmployee extends Employee {  
    int OTHrs;  
  
    public ParttimeEmployee(String newname, double newsalary, int newOTHrs) {  
        super(newname, newsalary);  
        OTHrs=newOTHrs;  
    }  
  
    public void displayEmpInfo() {  
        displayInfo();  
        System.out.println("OT Hours: " +OTHrs);  
    }  
}
```

OUTPUT:

```
Employee name: Alice  
Employee Salary: 20000.0  
OT Hours: 7
```

```
Employee name: John  
Employee Salary: 34000.0  
OT Hours: 2
```

Q2

Question:

Create a Java class named "**Student**" with the attributes (String) name, and (int) age. Write an overloaded constructor to initialize the attributes and provide getters for both attributes. Implement a method named "**displayInfo**" that displays the details of a student. Create another class called "**Course**" which is a child of Student class with an attribute (String) courseName. Create an overloaded constructor which supports the following object creation. Implement a method called "**displayCourseInfo**" to display all the details of the course including student name, age, and course name.

```
Course course = new Course("Alice", 20, "Java Programming");
```

Create a class called "**StudentApp**" with the main method. Create 2 Course objects and display the details by calling the **displayCourseInfo** method.

Student class

```
public class Student {  
  
    String name;  
    int age;  
  
    public Student(String newname,int newage) {  
        name=newname;  
        age=newage;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
  
    public void setAge(int age) {  
        this.age = age;  
    }  
  
    void displayInfo() {  
        System.out.println("Student name: "+getName());  
        System.out.println("Student age: "+getAge());  
    }  
}
```

Course class

```
public class Course extends Student{

    String courseName;

    public Course (String newname,int newage,String newcourseName) {
        super(newname,newage);
        courseName=newcourseName;
    }

    void displayCourseInfo() {
        displayInfo();
        System.out.println("Course name: " +courseName);
    }

}
```

OUTPUT:

```
Student name: Alice
Student age: 20
Course name: Java Programming
```

Q3

Question:

Create a class "Vehicle" with attributes (String) model, and (double) rentalPricePerDay. Create an overloaded constructor to assign the values to the attributes. Implement a method called "calculateRentalCost" to calculate the rental cost for a given number of days using the below method signature.

```
public double calculateRentalCost(int days);
```

Create 2 classes called "Car" and "Motorcycle" which are derived from the Vehicle class. Cars have an additional attribute (int) numSeats and motorcycles have an additional attribute (int) engineCapacity. Add overloaded constructors to each child class which accepts all 3 attributes.

Write a program that creates instances of both cars and motorcycles, calculates the rental cost for a specified number of days, and displays the total cost.

Vehicle class

```
public class Vehicle {  
  
    String model;  
    double rentalPricePerDay;  
  
    public Vehicle(String newmodel, double newrentalPricePerDay) {  
  
        model=newmodel;  
        rentalPricePerDay=newrentalPricePerDay;  
  
    }  
  
    public double calculateRentalCost(int days) {  
        return rentalPricePerDay * (days);  
    }  
  
}
```

Car class

```
public class Car extends Vehicle{
    int numSeats;

    public Car(String newmodel,double newrentalPricePerDay,int newnumSeats) {
        super(newmodel,newrentalPricePerDay);
        numSeats=newnumSeats;
    }
}
```

Motorcycle class

```
public class Motorcycle extends Vehicle{
    int engineCapacity;

    public Motorcycle(String newmodel,double newrentalPricePerDay,int newengineCapacity) {
        super(newmodel,newrentalPricePerDay);
        engineCapacity=newengineCapacity;
    }
}
```

OUTPUT:

```
Rent a Car
Model: Prius
No.of days: 3
Rent per Day :7500.0
Total cost : $22500.0

Rent a Bike
Model: Honet
No.of days: 3
Rent per Day :2000.0
Total cost : $6000.0
```

Q4

Assignment 01 - Online Test 01

Duration - 45 Minutes

Question:

Create a class called **"Product"** with attributes (int) productId, (String) name, and (double) price. Include an overloaded constructor to initialize these attributes. Create a getter and a setter to productId. Implement a method called **"displaydetails"** to print the values of above attributes. Create a class called **"Electronics"** that extends the Product class. Add additional attributes as (String) brand, and (String) model. Create a **"Clothing"** class that extends the Product class. Add (String) size and (String) color as specific attributes to Clothing class. Implement an overloaded constructor for each class (Clothing and Electronics). Implement a method called **"display"** in both classes to display all the details of each object. (Hint: display() in Clothing class should display productId, name, price, size and color. display() in Electronics class should display productId, name, price, brand and model)

Create another class called **"ShoppingCart"** with main method. Create 2 objects from (1 from each) Clothing and Electronics class and display the details by calling the appropriate methods.

Product class


```

public class Product {

    int productId;
    String name;
    double price;

    public Product (int newproductId, String newname, double newprice) {

        productId=newproductId;
        name=newname;
        price=newprice;

    }

    public int getProductId() {
        return productId;
    }

    public void setProductId(int productId) {
        this.productId = productId;
    }

    public void displaydetails() {
        System.out.println("Product ID: " +getProductId());
        System.out.println("Product Name: " +name);
        System.out.println("Product Price: " +price);

    }

}

```

Clothing class

```

public class Clothing extends Product {

    String size;
    String color;

    public Clothing(int newproductId, String newname, double newprice, String newsize, String newcolor) {

        super(newproductId, newname, newprice);

        size=newsize;
        color=newcolor;

    }

    public void displayInfo() {

        displaydetails();
        System.out.println("Size: "+size);
        System.out.println("Color: "+color);

    }

}

```

Electronics class

```

public class Electronics extends Product {

    String brand;
    String model;

    public Electronics (int newproductId, String newname, double newprice, String newbrand, String newmodel) {

        super(newproductId, newname, newprice);
        brand=newbrand;
        model=newmodel;

    }

    public void displayInfo() {

        displaydetails();
        System.out.println("Brand: "+brand);
        System.out.println("Model: "+model);

    }

}

```

ShoppingCart class

```
public class ShoppingCart {  
  
    public static void main(String[] args) {  
  
        Clothing c1 = new Clothing (18,"Polo Shirt",4578.00,"M","red");  
        Electronics e1 = new Electronics (34,"Plug socket",20000.00,"Orange","Safety model");  
  
        c1.displayInfo();  
        System.out.println();  
        e1.displayInfo();  
  
    }  
}
```

OUTPUT:

```
Product ID: 18  
Product Name: Polo Shirt  
Product Price: 4578.0  
Size: M  
Color: red  
  
Product ID: 34  
Product Name: Plug socket  
Product Price: 20000.0  
Brand: Orange  
Model: Safety model
```

Q5

Duration - 45 Minutes

Question:

Create a Java class named "**TemperatureConverter**" with an attribute (double) temperature. Add a default constructor that sets the attribute to 0 and a getter method. Implement a method named "celsiusToFahrenheit" that takes a double value representing a temperature in Celsius as a parameter and returns the equivalent temperature in Fahrenheit. Add another method named "fahrenheitToCelsius" that takes a double value representing a temperature in Fahrenheit as a parameter and returns the equivalent temperature in Celsius.

Create a class called "**TempApp**" with the main method. Create 2 TemperatureConverter objects and accept the user inputs for attribute (one as Celsius and one as Fahrenheit) and assign them. Call the appropriate methods and print the values.

Sample Output

```
Enter the temperature in celsius : 25
Enter the temperature in fahrenheit : 122
25.0 Celsius is equal to 77.0 Fahrenheit
122.0 Fahrenheit is equal to 50.0 Celsius
```

TemperatureConverter class

```
public class TemperatureConverter {  
    double temperature;  
  
    public TemperatureConverter() {  
        temperature=0.0;  
    }  
  
    public double getTemperature() {  
        return temperature;  
    }  
  
    public double celciusToFahrenheit(double celcius) {  
        return (celcius * 9/5) + 32;  
    }  
  
    public double fahrenheitToCelcius(double fahrenheit) {  
        return (fahrenheit -32) * 5/9;  
    }  
  
}
```

TempApp clasS

```

import java.util.Scanner;

public class TempApp {

    public static void main(String[] args) {
        TemperatureConverter tc1 = new TemperatureConverter();
        TemperatureConverter tc2 = new TemperatureConverter();
        Scanner myScanner = new Scanner(System.in);

        System.out.println("Enter the temperature in celcius : ");
        double celcius = myScanner.nextDouble();

        System.out.println("Enter the temperature in Fahrenheit : ");
        double fahrenheit = myScanner.nextDouble();

        double farfromcel = tc1.celciusToFarenheit(celcius);
        System.out.println(celcius + "Celcius is equal to " + farfromcel+ "Fahrenheit");

        double celfromfar = tc2.fahrenheitToCelcius(fahrenheit);
        System.out.println(fahrenheit + "Fahrenheit is equal to " + celfromfar+ "Celsius");

        myScanner.close();

    }
}

```

OUTPUT:

```

Enter the temperature in celcius :
25
Enter the temperature in Fahrenheit :
122
25.0Celcius is equal to 77.0Fahrenheit
122.0Fahrenheit is equal to 50.0Celsius

```

Q6

Duration - 45 minutes

Question:

✓ Create a Java class called "MathOperations" with following methods.

Implement a method that takes two integers as input parameters and prints the multiplication table for the first input parameter up to the second parameter.

```
public void drawMultiplicationTable (int number, int range);
```

Implement another method that reads a positive integer (n) from the user and calculates the factorial of that number(n!) using a loop and return the value.

```
public int factorial ();
```

Create another class called "MathApp" with main method and call both the implemented methods and display the outputs properly.

MathOperations class

```

import java.util.Scanner;
public class MathOperations {

    public void drawMultiplicationTable(int number,int range) {
        for (int i=1;i<=range;i++) {
            System.out.println(number + "x" + i +"=" +(number*i));
        }
    }

    public int factorial() {

        Scanner myScanner = new Scanner(System.in);
        System.out.println("Enter a positive integer: ");
        int n = myScanner.nextInt();

        int result = 1;
        for(int i=1; i<=n;i++) {
            result *=i;
        }

        return result;
    }
}

```

MathApp class

```

public class MathApp {

    public static void main(String[] args) {

        MathOperations mops = new MathOperations();

        mops.drawMultiplicationTable(5, 10);

        int factorialResult = mops.factorial();
        System.out.println("The factorial is: "+factorialResult);

    }

}

```

OUTPUT:


```
5x1=5
5x2=10
5x3=15
5x4=20
5x5=25
5x6=30
5x7=35
5x8=40
5x9=45
5x10=50
Enter a positive integer:
5
The factorial is: 120
```