

Assignment-1

Name: Chebrolu Rukmini

College: SRM UNIVERSITY AP

Q1: Create a class named “Order” that performs order processing of a single item. It has following fields: customer name, customer number, quantity ordered, unit price. Include setter and getter methods for each field. The setter methods prompt the user for values for each field. Create a method (calculatePrice()) to calculate the total price (quantity x unit price) and another method to display total price calculated. Create a subclass named “ShippedOrder” that overrides calculatePrice () by adding a shipping and handling charge of Rs 4.00 per item. Write an application named “UseOrder” that instantiates an object of each of classe and display the results separately.

Solution

Order.java

```
package q1;
public class Order
{
    private String CustomerName;
    private int CustomerNumber;
    private int QuantityOrdered;
    private double UnitPrice;
    protected double TotalPrice;

    public void setCustomerNumber(int num)
    {
        CustomerNumber=num;
    }

    public int getCustomberNumber()
    {
        return CustomerNumber;
    }

    public void setQuantityOrdered(int order)
    {
        QuantityOrdered=order;
    }

    public int getQuantityOrdered()
    {
        return QuantityOrdered;
    }

    public void setUnitPrice(double price)
```

```

    {
        UnitPrice=price;
    }

    public double getUnitPrice()
    {
        return UnitPrice;
    }

    public void setCustomerName(String name)
    {
        CustomerName=name;
    }

    public String getCustomerName()
    {
        return CustomerName;
    }

    public void inputCustomerName()
    {
        String customername = new String ("");
        System.out.println("Please enter customer's name :");
    }

    public void displayCustomerName()
    {
        System.out.println("The customer name you entered is : " + CustomerName);
    }

    public void inputCustomerNumber()
    {
        String customernumber = new String ("");
        System.out.println("Please enter customer's number :");
    }

    public void displayCustomerNumber()
    {
        System.out.println("The customer number you entered is : " + CustomerNumber);
    }

    public void inputQuantityOrdered()
    {
        String quantityordered = new String ("");
        System.out.println("Please enter the quantity you want: ");
    }

```

```

    }

    public void displayQuantityOrdered()
    {
        System.out.println("The quantity you ordered is : " + QuantityOrdered);
    }

    public void inputUnitPrice()
    {
        String unitprice=new String ("");
        System.out.println("Please enter the price per unit:");
    }

    public void displayUnitPrice()
    {
        System.out.println("The price per unit is: " + UnitPrice);
    }

    public void TotalPrice()
    {
        TotalPrice=QuantityOrdered*UnitPrice;
    }

    public void displayTotalPrice()
    {
        System.out.println("The total price is: " + TotalPrice);
    }

    public void ItemOrder()
    {
        String ItemOrdered = new String (" ");
        System.out.println(" Enter the item you want to order:");
    }

    public void choice(){
        String unitprice=new String ("");
        System.out.println("Enter the item you want to order:");
        String choice = null;
        int OrderChoice = Integer.parseInt(choice);
    }
}

```

ShippedOrder.java

```
package q1;

public class ShippedOrder extends Order
{
    private int TotalCost;
    int ShippingCost;
    char OrderChoice;

    public ShippedOrder(int i) {
        // TODO Auto-generated constructor stub
    }

    public void ShippedOrderChoice() {
        String choice = new String ();
        System.out.println("You want to ship your order? Y or N");

        String Y = null;
        String N;
        if(choice==Y)
            ShippingCost = 4;

        else
            ShippingCost = 0;

        TotalCost = TotalCost + ShippingCost;
    }
}
```

UseOrder.java

```
package q1;

public class UseOrder {
    public static void main (String [] args)
    {

        Order s = new ShippedOrder(1);
        s.displayTotalPrice();
        ShippedOrder s2 = new ShippedOrder(5);
        s2.displayTotalPrice();
    }
}
```

Q2. a) Create a class named Year that contains a data that holds the number of days in a year. Make a constructor that sets the number of days to 365. Include a getter method that displays the number of days. Create another subclass named LeapYear. It calls Year's constructor and sets the number of days to 366.

Make a service class name UseYear that instantiates one object of each class and displays their data. Save the files as Year.java, LeapYear.java, and UseYear.java

Year.java

package q2;

```
public class Year {
    public int num_of_days;

    public Year() {
        num_of_days=365;
    }

    public int getNum_of_days() {
        return num_of_days;
    }

}
```

LeapYear.java

package q2;

```
public class LeapYear extends Year{

    public LeapYear() {
        num_of_days=366;
    }

}
```

UseYear.java

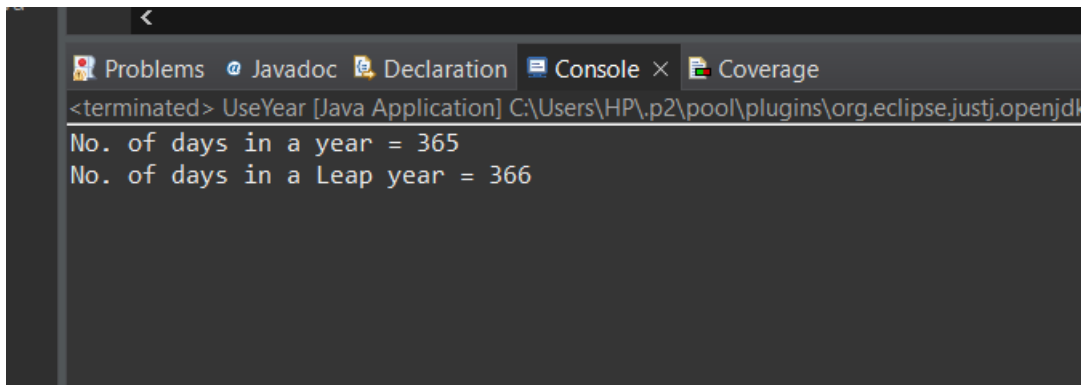
package q2;

```
public class UseYear {

    public static void main(String[] args) {
        Year year = new Year();
        LeapYear leapyear = new LeapYear();

        System.out.println("No. of days in a year = "+year.getNum_of_days());
        System.out.println("No. of days in a Leap year = "+leapyear.getNum_of_days());
    }

}
```



```
<terminated> UseYear [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk
```

```
No. of days in a year = 365
No. of days in a Leap year = 366
```

Q2(b) Add a method named `daysElapsed()` to the `Year` class you created in last Exercise 2(a). The `daysElapsed()` method accepts two arguments representing a month and a day; it returns an integer indicating the number of days that have elapsed since January 1 of that year. For example, on March 3 in nonleap years, 61 days have elapsed (31 in January, 28 in February, and 2 in March). Create a `daysElapsed()` method for the `LeapYear` class that overrides the method in the `Year` class. For example, on March 3 in a `LeapYear`, 62 days have elapsed (31 in January, 29 in February, and 2 in March). Write an application named `UseYear2` that prompts the user for a month and day, and calculates the days elapsed in a `Year` and in a `LeapYear`. Save the files as `Year2.java`, `LeapYear2.java`, and `UseYear2.java`.

Year2.java

```
package q2b;
```

```
public class Year2 {
    public int num_of_days;

    public Year2() {
        num_of_days=365;
    }

    public int getNum_of_days() {
        return num_of_days;
    }

    public int daysElapsed(int day,int month)
    {
        int temp = 0;
        for(int i=1;i<month;i++)
        {
            if(i==1 || i==3 || i==5 || i==7 || i==8 || i==10 || i==12)
            {
                temp += 31;
            }
            else if(i==4 || i==6 || i==9 || i==11)
            {
                temp += 30;
            }
            else if(i==2)
            {
```

```
temp += 28;
}
}
temp += day;
return temp;
}
}
```

LeapYear2.java

```
package q2b;

public class LeapYear2 extends Year2{

    public LeapYear2() {
        num_of_days=366;
    }
    public int daysElapsed(int day,int month)
    {
        int temp = 0;
        for(int i=1;i<month;i++)
        {
            if(i==1 || i==3 || i==5 || i==7 || i==8 || i==10 || i==12)
            {
                temp += 31;
            }
            else if(i==4 || i==6 || i==9 || i==11)
            {
                temp += 30;
            }
            else if(i==2)
            {
                temp += 29;
            }
        }
        temp += day;
        return temp;
    }

}
```

UseYear2.java

```
package q2b;

public class UseYear2 {

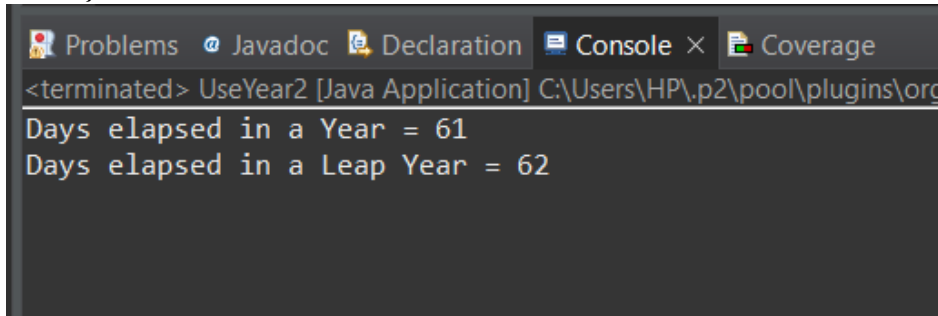
    public static void main(String[] args) {
        int day = 2,month = 3;
```

```

        Year2 year = new Year2();
        LeapYear2 leapyear = new LeapYear2();

        System.out.println("Days elapsed in a Year =
"+year.daysElapsed(day,month));
        System.out.println("Days elapsed in a Leap Year =
"+leapyear.daysElapsed(day,month));
    }
}

```



```

<terminated> UseYear2 [Java Application] C:\Users\HP\p2\pool\plugins\org
Days elapsed in a Year = 61
Days elapsed in a Leap Year = 62

```

Q3 Create a class named HotelRoom, having roomNumber as integer and nightlyRentCharge as double.

Include getter methods for required field and a constructor that requires an integer argument representing the room number.

The constructor sets the room rate based on the room number; rooms numbered 299 and below are Rs 6900 per night, and others are Rs 8900 per night.

Create an extended class named Suite whose constructor requires a room number and adds a Rs 400 surcharge

to the regular hotel room rate, which again is based on the room number.

Write an application named HotelRoomService that creates an object of each class, and demonstrate that all the methods work correctly

HotelRoom.java

```

class HotelRoom {
    final double rate1 = 6900;
    final double rate2 = 8900;
    final int regularrate = 400;

```

```

    private int roomNumber;
    private double nightlyRentCharge;

```

```

    HotelRoom (int Room ) {
        roomNumber=Room;
        if (roomNumber <= 299)//regularrate
            nightlyRentCharge = rate1;
        else

```



```

        nightlyRentCharge = rate2;
    }
    public int getRoom(){
        return roomNumber;
    }
    public double getRentalRate(){
        return nightlyRentCharge;
    }
    public void setRoom(int Room){
        roomNumber=Room;
    }
    public void setRentalRate(double nightlyRentCharge){

    }
}

```

Suite.java

```

class Suite extends HotelRoom {
    final double surcharge = 400.00;
    Suite (int roomNumber){
        super(roomNumber);
        setRentalRate(getRentalRate() + surcharge);
    }
}

```

HotelRoomService.java

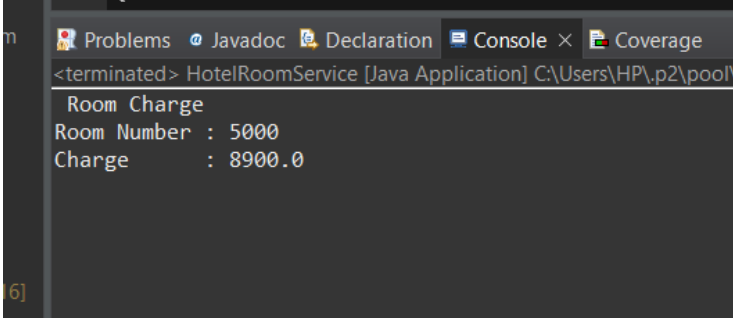
```

public class HotelRoomService {

    public static void main(String[] args){
        HotelRoom aRoom = new HotelRoom(5000);
        System.out.println(" Room Charge ");
        System.out.println("Room Number : " + aRoom.getRoom());
        System.out.println("Charge      : " + aRoom.getRentalRate());
    }
}

```

Output:



```

<terminated> HotelRoomService [Java Application] C:\Users\HP\p2\pool\
Room Charge
Room Number : 5000
Charge      : 8900.0

```

Q4. Create a class named Vehicle that acts as a superclass for vehicle types. The Vehicle class contains private variables for the number of wheels and the average number of miles per gallon. It also contains a constructor with integer arguments for the number of wheels and average miles per gallon, and a toString() method that returns a String containing these values. Create two subclasses, Car and Motorcycle, that extend the Vehicle class. Each subclass contains a constructor that accepts the miles-per-gallon value as an argument and forces the number of wheels to the appropriate value—2 for a Motorcycle and 4 for a Car. Write a UseVehicle class to instantiate the two Vehicle objects and print the objects' values.

*/

```
package q4;

public class Vehicle {

    private int no_of_wheels;
    private int miles_per_gallon;

    Vehicle (int no_of_wheels, int miles_per_gallon) {
        this.no_of_wheels=no_of_wheels;
        this.miles_per_gallon=miles_per_gallon;
    }

    @Override
    public String toString() {
        return "no_of_wheels=" + no_of_wheels + ", miles_per_gallon=" +
miles_per_gallon ;
    }

    public void display() {
        System.out.println("Wheels: " + toString() + " Miles Per Gallon: " +toString());
    }

}-----
```

```
package q4;

public class Car extends Vehicle {
    Car (int CarMilesPerGallon) {

        super(4, CarMilesPerGallon);
    }
}
```

```
package q4;

public class Motorcycle extends Vehicle{
    Motorcycle (int MotorcycleMilesPerGallon) {

        super(2, MotorcycleMilesPerGallon);
    }
}
```

```

    }

}

package q4;

public class UseVehicle {
    public static void main(String[] args) {
        Car car = new Car(30);
        Motorcycle motorcycle = new Motorcycle(60);

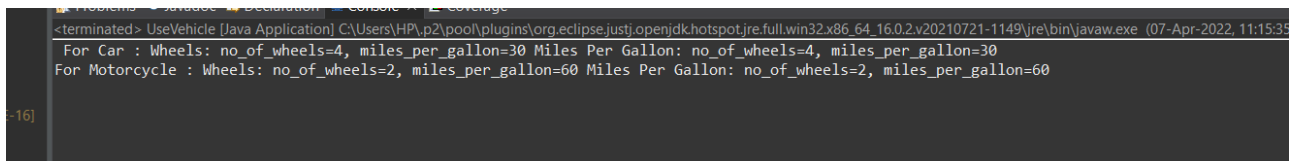
        System.out.print(" For Car : ");
        car.display();
        System.out.print("For Motorcycle : ");
        motorcycle.display();

    }

}

```

Output:



```

<terminated> UseVehicle [Java Application] C:\Users\HP\AppData\Local\Temp\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (07-Apr-2022, 11:15:35)
For Car : Wheels: no_of_wheels=4, miles_per_gallon=30 Miles Per Gallon: no_of_wheels=4, miles_per_gallon=30
For Motorcycle : Wheels: no_of_wheels=2, miles_per_gallon=60 Miles Per Gallon: no_of_wheels=2, miles_per_gallon=60

```

Q5. Create a class named CollegeCourse that includes data fields that hold the department (for example, “ENG”), the course number (for example, 101), the credits (for example, 3), and the fee for the course (for example, Rs360). All of the fields are required as arguments to the constructor, except for the fee, which is calculated at \$120 per credit hour. Add a display() method that displays the course data. Create a subclass named LabCourse that adds \$50 to the course fee. Override the parent class display() method to indicate that the course is a lab course and to display all the data. Write an application named UseCourse that prompts the user for course information. If the user enters a class in any of the following departments, create a LabCourse: BIO, CHM, CIS, or PHY. If the user enters any other department, create a CollegeCourse that does not include the lab fee. Then display the course data. Save the files as CollegeCourse.java, LabCourse.java, and UseCourse.java

```

package q5;
import java.util.Scanner;
public class CollegeCourse {
    protected String dept;
    protected int course_num;
    protected int credits;
    private double fee;

    public CollegeCourse(String dept, int course_num, int credits) {

```

```

        super();
        this.dept = dept;
        this.course_num=course_num;
        this.credits = credits;
        this.fee = this.credits*120;
    }

    public void display() {
        System.out.println(this.dept+""+this.course_num);
        System.out.println("No Lab Course");
        System.out.println(this.credits+"credits");
        System.out.println("Total Amount of fee is $" +this.fee);
    }

}

```

```

package q5;

```

```

public class LabCourse extends CollegeCourse{
    private int LabFee;

    public LabCourse(String dept, int course_num, int credits, int LabFee) {
        super(dept, course_num, credits);
        this.LabFee=LabFee;
        // TODO Auto-generated constructor stub
    }

    public void display() {
        System.out.println(this.dept+""+this.course_num);
        System.out.println("Lab Course");
        System.out.println(this.credits+"credits");
        System.out.println("Lab Fee is $" +this.LabFee);
        System.out.println("Total Amount of fee is $ " +this.LabFee);
    }

}

```

```

package q5;
import java.util.*;
public class USeCourse {

    public static void main(String[] args) {
        Scanner input=new Scanner(System.in);
        String dept;
        String[] labCourses= {"CHM","PHY","BIO"};
        int course_num,credits,found=0,i;
        System.out.print("Enter a course :");
        dept=input.nextLine();
        System.out.print("Enter Course Number");
        course_num=input.nextInt();
    }
}

```

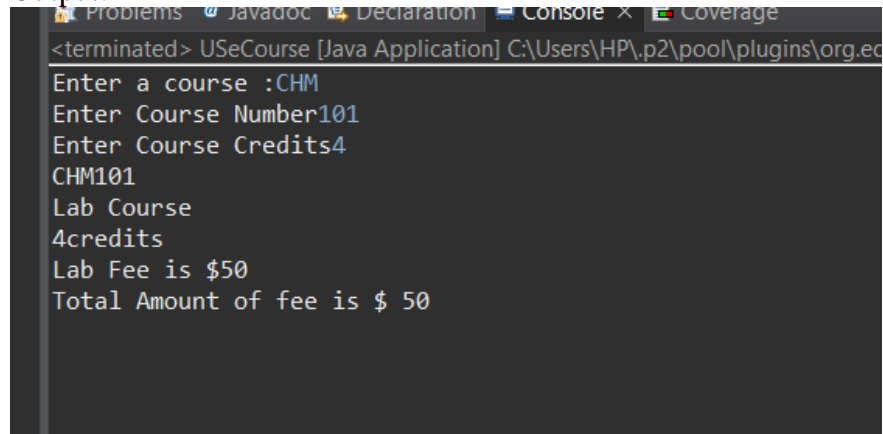
```

System.out.print("Enter Course Credits");
credits=input.nextInt();
for(i=0;i<labCourses.length;i++) {
    if(labCourses[i].equals(dept)) {
        found=1;
    }
}
if(found==1) {
    LabCourse l=new LabCourse(dept,course_num,credits,50);
    l.display();
}
else {
    CollegeCourse c=new CollegeCourse(dept,course_num,credits);
    c.display();
}
}

```

}

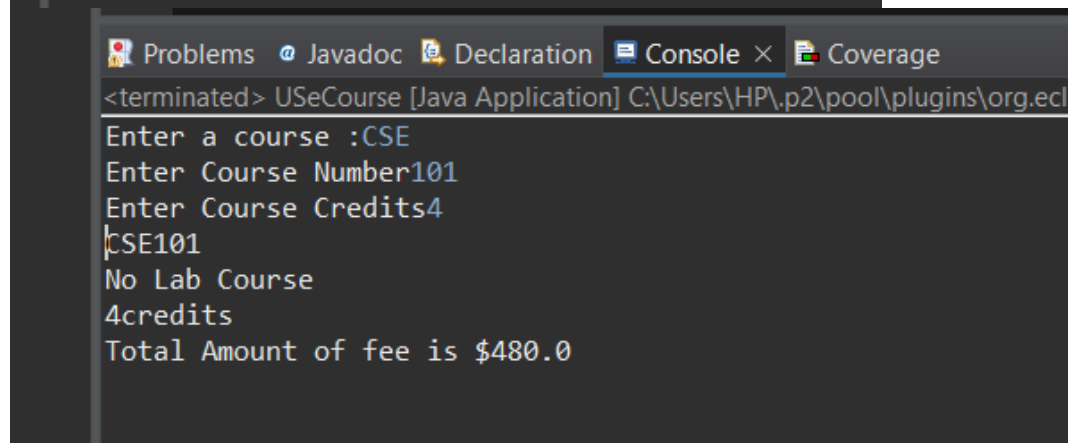
Output:



```

<terminated> USeCourse [Java Application] C:\Users\HP\.p2\pool\plugins\org.ec
Enter a course :CHM
Enter Course Number101
Enter Course Credits4
CHM101
Lab Course
4credits
Lab Fee is $50
Total Amount of fee is $ 50

```



```

Problems Javadoc Declaration Console Coverage
<terminated> USeCourse [Java Application] C:\Users\HP\.p2\pool\plugins\org.ec
Enter a course :CSE
Enter Course Number101
Enter Course Credits4
CSE101
No Lab Course
4credits
Total Amount of fee is $480.0

```