CSE2005 (OOP in Java) Lab-3 01/12/2021

KHAN MOHD. OWAIS RAZA
20BCD7138

Q.1] Mick's Wicks makes candles in various sizes. Create a class for the business named Candle that contains data fields for color, height, and price. Create get methods for all three fields. Create set methods for color and height, but not for price. Instead, when height is set, determine the price as $2 per inch. Create a child class named ScentedCandle that contains an additional data field named scent and methods to get and set it. In the child class, override the parent's setHeight() method to set the price of a ScentedCandle object at $3 per inch. Write an application that instantiates an object of each type and displays the details. Save the files as Candle.java, ScentedCandle.java, and DemoCandles.java.

DemoCandles.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-3*/
public class DemoCandles
{
  public static void main(String args[])
  {
    Candle aCandle = new Candle();
    ScentedCandle aScentedCandle = new ScentedCandle();
    aCandle.setColor("pink");
    aCandle.setHeight(6);
    aScentedCandle.setColor("white");
    aScentedCandle.setScent("gardinia");
    aScentedCandle.setHeight(6);
    System.out.println("The"
                  + aCandle.getHeight() +
                  "inch"
                  + aCandle.getColor() +
                  "candle costs $"
                  + aCandle.getPrice());
    System.out.println("The"
                  + aScentedCandle.getHeight() +
                  "inch"
                  + aScentedCandle.getScent() +
                  "  " + aScentedCandle.getColor() +
                  "candle costs $"
                  + aScentedCandle.getPrice());
    }
}
```

ScentedCandle.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-3*/
public class ScentedCandle extends Candle
{
  private String scent;
  public String getScent()
  {
   return scent;
  }
  public void setScent(String scent)
  {
    this.scent = scent;
  }
  @Override
  public void setHeight(int h)
  {
    final double PER_INCH = 3;
    super.setHeight(h);
    price = h * PER_INCH;
  }
}
```

Candle.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-3*/
public class Candle
{
  private String color;
  private int height;
  protected double price;
  public String getColor()
  {
    return color;
  }
  public int getHeight()
  {
    return height;
  }
  public double getPrice()
  {
    return price;
  }
  public void setColor(String c)
  {
    color = c;
  }
  public void setHeight(int h)
  {
    final double PER_INCH = 2;
    height = h;
    price = height * PER_INCH;
  }
}
```

Output :–

```
Output

$ The 6 inch pink candle costs $12.0
  The 6 inch gardinia white candle costs $18.0
```

Q.2] Create a class named Poem that contains fields for the name of the poem and the number of lines in it. Include a constructor that requires values for both fields. Also include get methods to retrieve field values. Create three subclasses: Couplet, Limerick, and Haiku. The constructor for each subclass requires only a title; the lines field is set using a constant value. A couplet has two lines, a limerick has five lines, and a haiku has three lines. Create an application that demonstrates usage of an object of each type. Save the files as Poem.java, Couplet.java, Limerick.java, Haiku.java, and DemoPoems.java.

Poem.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
public class Poem {
    String title;
    int lines;
    public Poem(String title, int lines) {
        this.title = title;
        this.lines = lines;
    }
    public String getTitle() {
        return title;
    }
    public int getLines() {
        return lines;
    }

}
```

Couplet.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
public class Couplet extends Poem
{
  public Couplet(String title)
     {
       super(title,2);
    }
}
```

Limerick :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
public class Limerick extends Poem {
    public Limerick(String title) {
        super(title,5);
    }
}
```

Haiku.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
public class Haiku extends Poem
{
  public Haiku(String title)
  {
    super(title,3);
    }
}
```

DemoPoems.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
import java.util.*;
public class DemoPoems
{
    public static void main(String[] args)
    {
        Poem poem1 = new Poem("The Raven", 84);
        Couplet poem2 = new Couplet("True Wit");
        Limerick poem3 = new Limerick("There was an Old Man with a Beard");
        Haiku poem4 = new Haiku("The Wren");
        display(poem1);
        display(poem2);
        display(poem3);
        display(poem4);
    }
    public static void display(Poem p)
    {
        System.out.println(" Poem: "
                           + p.getTitle() +
                           " Lines: "
                           + p.getLines());
    }
}
```

Output :–

```
Output

$ Number of lines in Couplet : 2
  Title : True Wit
  Number of lines in Limerick : 5
  Title : There was an Old Man with a Beard
  Number of lines in Haiku : 3
  Title : The Wren
```

Q.3] 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, $360). All of the fields are required as arguments to the constructor, except for the fee, which is calculated at $120 per credit hour. Include 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.

CollegeCourse :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
public class CollegeCourse {
   String dept;
   int id;
   double credits;
   double price;
   public CollegeCourse(String depart, int id, double cred)
   {
        this.dept = depart;
        this.id = id;
        this.credits = cred;
        price = credits * 120;
    }
    public void display(){
        System.out.println(dept+id+
                           "\nNon-lab course\n"
                           + credits +
                           " credits\nTotal fee is $"
                           + price);
    }
}
```

LabCourse.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
public class LabCourse extends CollegeCourse
{
   public LabCourse(String dept, int id, double credits)
   {
      super(dept, id, credits);
      price+= 50;
   }
    public void display()
    {
       System.out.println(dept + id + "\n"+
                          "Lab course\n"
                          + credits +
                          "credits\nLab fee is $50\nTotal fee is $"
                          + price);
    }
}
```

UseCourse.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
import java.util.Scanner;
public class UseCourse
{
   public static void main(String[] args)
   {
      String dept;
      int id;
      double credits;
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter the course code : ");
      dept = sc.nextLine().toUpperCase();
      System.out.print("Enter the course ID : ");
      id = sc.nextInt();
      sc.nextLine();
      System.out.print("Enter the credits : ");
      credits = sc.nextDouble();
      sc.nextLine();
      System.out.println();
      switch(dept)
      {
         case "BIO":
         case "CHM":
         case "CIS":
         case "PHY":
       LabCourse labCourse = new LabCourse(dept, id, credits);
       labCourse.display();
       break;
       default:
       CollegeCourse colCourse = new CollegeCourse(dept, id, credits);
       colCourse.display();
       break;
        }
    }
}
```

Output :–

```
Output
$ Enter the course code : BIO
  Enter the course ID : 201
  Enter the credits : 4
  BIO201
  Lab course
  4.0 credits
  Lab fee is $50
  Total fee is $530.0
```

Q.4] Develop a set of classes for a college to use in various student service and personnel applications. Classes you need to design include the following:

• Person - A Person contains a first name, last name, street address, zip code, and phone number. The class also includes a method that sets each data field, using a series of dialog boxes and a display method that displays all of a Person's information on a single line at the command line on the screen.

• CollegeEmployee - CollegeEmployee descends from Person. A CollegeEmployee also includes a Social Security number, an annual salary, and a department name, as well as methods that override the Person methods to accept and display all CollegeEmployee data.

• Faculty - Faculty descends from CollegeEmployee. This class also includes a Boolean field that indicates whether the Faculty member is tenured, as well as methods that override the CollegeEmployee methods to accept and display this additional piece of information.

• Student - Student descends from Person. In addition to the fields available in Person, a Student contains a major field of study and a grade point average as well as methods that override the Person methods to accept and display these additional facts.

Write an application named CollegeList that declares an array of four "regular" CollegeEmployees, three Faculty, and seven Students. Prompt the user to specify which type of person's data will be entered (C, F, or S), or allow the user to quit (Q). While the user chooses to continue (that is, does not quit), accept data entry for the appropriate type of Person.

If the user attempts to enter data for more than four CollegeEmployees, three Faculty, or seven Students, display an error message.

When the user quits, display a report on the screen listing each group of Persons under the appropriate heading of "College Employees," "Faculty," or "Students." If the user has not entered data for one or more types of Persons during a session, display an appropriate message under the appropriate heading.

Save the files as Person.java, CollegeEmployee.java, Faculty.java, Student.java, and CollegeList.java.

Person.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138*/
/* CSE2005 LAB-3 */
import java.util.*;
public class Person
{
    private String firstName;
    private String lastName;
    public String address;
    private String zip;
    private String phone;
    Scanner sc = new Scanner(System.in);
    public void setData()
    {
        System.out.println("Enter the first name: ");
        firstName = sc.nextLine();

        System.out.println("Enter the last name: ");
        lastName = sc.nextLine();

        System.out.println("Enter address: ");
        address = sc.nextLine();

        System.out.println("Enter zip code: ");
        zip = sc.nextLine();

        System.out.println("Enter the phone number: ");
        phone = sc.nextLine();
    }
    public void display()
    {
        System.out.println(firstName+" "+lastName+" "+address+" "+zip+" "+phone);
    }
}
```

CollegeEmployee.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138*/
/* CSE2005 LAB-3 */
import java.util.*;
public class CollegeEmployee extends Person
{
    protected String ssn;
    protected double annualSalary;
    protected String dept;
    Scanner sc = new Scanner(System.in);
    public CollegeEmployee()
    {
       super();
    }
    public void setData()
    {
        super.setData();
        System.out.println("Enter ssn: ");
        ssn = sc.nextLine();
        System.out.println("Enter annual Salary: ");
        annualSalary = sc.nextDouble();
        System.out.println("Enter dept: ");
        dept = sc.nextLine();
    }
    public void display()
    {
        super.display();
        System.out.println("SSN: "
                           + ssn +
                           " Salary $"
                           + annualSalary +
                           " Department: "
                           + dept);
    }
}
```

Faculty.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138*/
/* CSE2005 LAB-3 */
import java.util.*;
public class Faculty extends CollegeEmployee
{
    private boolean isTenured = false;
    Scanner sc = new Scanner(System.in);
    public Faculty()
    {
       super();
    }
    @Override
    public void setData() {
        String temp;
        super.setData();
        if(temp.charAt(0) == 'Y' || temp.charAt(0) == 'y')
            isTenured = true;
    }
    @Override
    public void display() {
        super.display();
        if(isTenured)
            System.out.println("Faculty member is tenured");
        else
            System.out.println("Faculty member is not tenured");
    }
}
```

Student.java :–

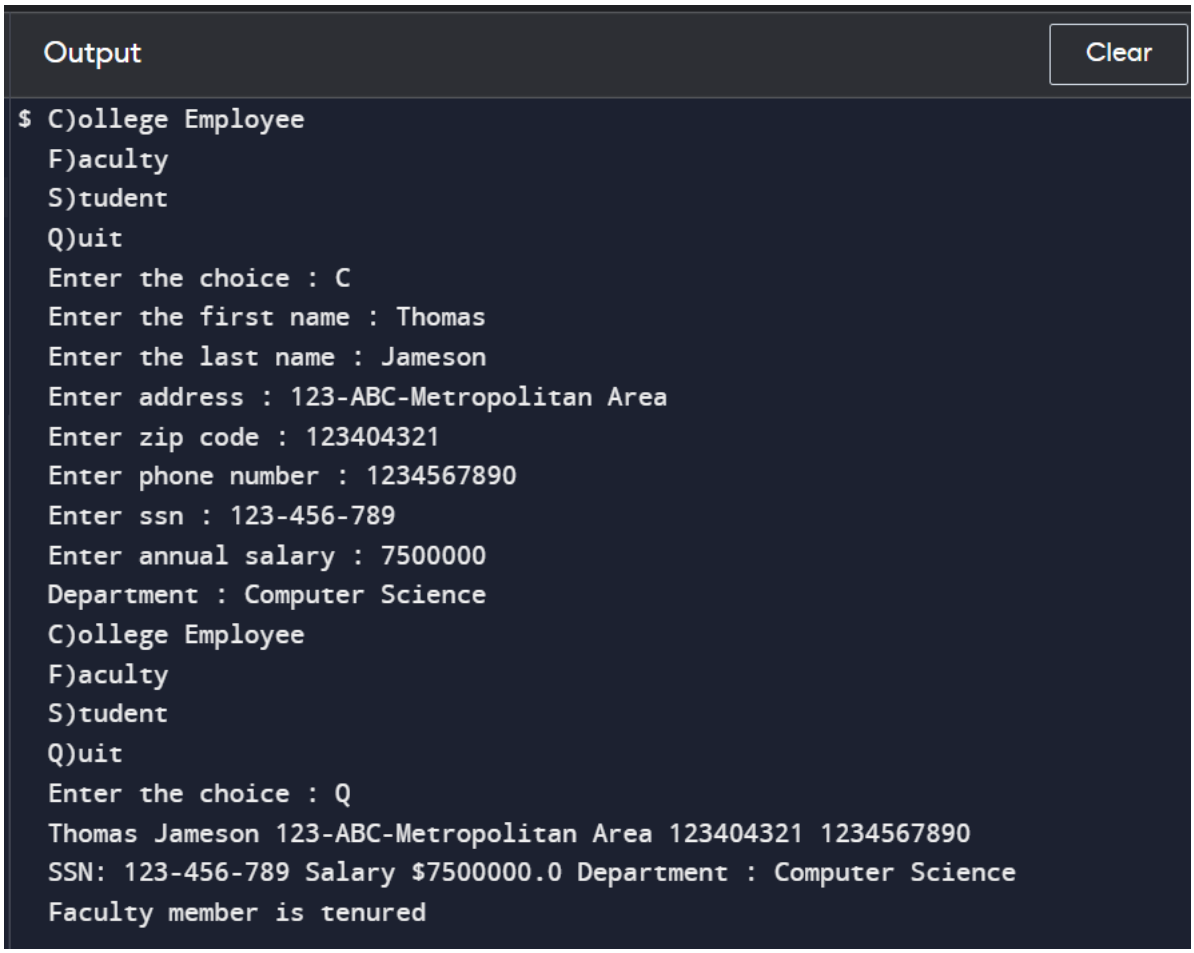
```
/* KHAN MOHD OWAIS RAZA 20BCD7138*/
/* CSE2005 LAB-3 */
import java.util.*;
public class Student extends Person
{
    private String major;
    private double gpa;
    Scanner sc = new Scanner(System.in);
    public Student()
    {
      super();
    }
    @Override
    public void setData() {
        super.setData();
        System.out.println("Please enter student's major");
        major = sc.nextLine();
        System.out.println("Please enter grade point average");
        gpa = sc.nextDouble();
    }
    @Override
    public void display() {
        super.display();
        System.out.println("Major: " + major + "  GPA: " + gpa);
    }
}
```

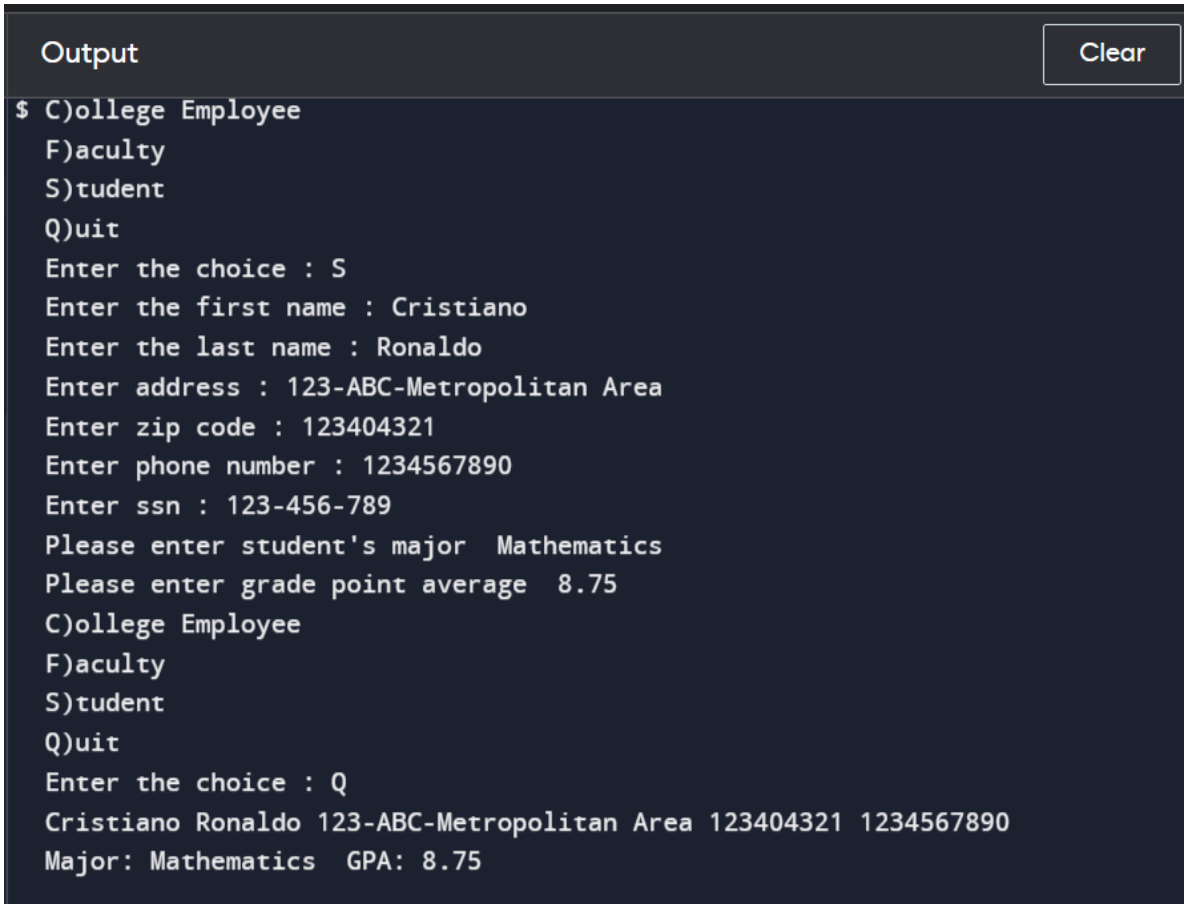
CollegeList.java :–

```
/* KHAN MOHD OWAIS RAZA 20BCD7138*/
/* CSE2005 LAB-3 */
import java.util.*;
public class CollegeList
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        CollegeEmployee[] emp = new CollegeEmployee[4];
        Faculty[] fac = new Faculty[3];
        Student[] stu = new Student[7];
        int empCount = 0, facCount = 0, stuCount = 0;
        char letter;
        String userInput;
        int x;
        System.out.println("Type C, F or S to enter data for" +
                           "\n(C)ollege employee\n(F)aculty\n(S)tudent" +
                           "\nor type Q to quit");
        userInput = input.next();
        letter = userInput.charAt(0);
        while (letter != 'Q')
        {
            if(letter == 'C')
            {
                if(empCount < emp.length)
                {
                    CollegeEmployee c = new CollegeEmployee();
                    c.setData();
                    emp[empCount] = c;
                    ++empCount;
                }
                else
                    System.out.println("Sorry - too many employee records have been entered");
            }
            else if(letter == 'F')
            {
                if(facCount < fac.length)
                {
                    Faculty f = new Faculty();
                    f.setData();
                    fac[facCount] = f;
                    ++facCount;
                }
                else
                    System.out.println("Sorry - too many faculty records have been entered");
            }
            else if(letter == 'S')
            {
                if(stuCount < stu.length)
                {
                    Student s = new Student();
                    s.setData();
                    stu[stuCount] = s;
                    ++stuCount;
                }
                else
                    System.out.println("Sorry - too many student records have been entered");
            }
            System.out.println("Type C, F or S to enter data for" +
                    "\n(C)ollege employee\n(F)aculty\n(S)tudent" +
                    "\nor type Q to quit");
            userInput = input.next();
            letter = userInput.charAt(0);
        }
```

```
        System.out.println("\nCollege Employees:");
        if(empCount == 0)
            System.out.println("No employees entered");
        else
            for(x = 0; x < empCount; ++x)
                emp[x].display();
        System.out.println("\nFaculty:");
        if(facCount == 0)
            System.out.println("No faculty entered");
        else
            for(x = 0; x < facCount; ++x)
                fac[x].display();
        System.out.println("\nStudents:");
        if(stuCount == 0)
            System.out.println("No students entered");
        else
            for(x = 0; x < stuCount; ++x)
                stu[x].display();
    }
}
```

Output :–




Output
Clear
$ C)ollege Employee
F)aculty
S)tudent
Q)uit
Enter the choice : S
Enter the first name : Cristiano
Enter the last name : Ronaldo
Enter address : 123-ABC-Metropolitan Area
Enter zip code : 123404321
Enter phone number : 1234567890
Enter ssn : 123-456-789
Please enter student's major Mathematics
Please enter grade point average 8.75
C)ollege Employee
F)aculty
S)tudent
Q)uit
Enter the choice : Q
Cristiano Ronaldo 123-ABC-Metropolitan Area 123404321 1234567890
Major: Mathematics GPA: 8.75