                                     Assignment-7

Name: JYOTIKRISHNA BEHERA

Roll: 16

Sec: C1

SIC: 20BCSB33

1.

class Person{

    private String name;

    private String address;

    Person(){}

    Person(String n, String a){

        name = n;

        address = a;

    }

    String getName(){

        return address;

    }

    String getAddress(){

        return address;

    }

    void setName(String n){

        name = n;

    }

    void setAddress(String a){

        address = a;

    }

    String tostring(){

        return name+"("+address+")";

    }

}

class studdent extends Person{

    private int numCourses;

    private String courses[];

    private int grades[];

    studdent(){}

    studdent(String n, String a){

        setName(n);

        setAddress(a);

    }

    void addCourseGrade(String c[], int g[]){

        numCourses = c.length;

        courses = c;

        grades = g;

    }

    void printGrades(){

        System.out.println("Number of courses = "+numCourses);

        for(int i=0;i<grades.length;i++)

        System.out.println((i+1)+":\t"+courses[i]+" :- "+grades[i]);

    }

    double getAverageGrade(){

        int sum = 0;

    for(int i=0;i<numCourses;i++)

        sum += grades[i];

        return 1.0\*sum/numCourses;

    }

    String tostring(){

        return "studdent: "+super.tostring();

    }

}

class Teacher extends Person{

    private int numCourses;

    private String courses[];

    Teacher(String n,String a){

    setName(n);

    setAddress(a);

}

boolean addCourse(String c){

    for(int i=0;i<numCourses;i++)

        if(courses[i]==c)

            return false;

    numCourses += 1;

    String temp[] = new String[numCourses];

    for(int i=0;i<numCourses-1;i++)

        temp[i] = courses[i];

    temp[numCourses-1] = c;

    courses = temp;

    return true;

}

boolean removeCourse(String c){

    for(int i=0;i<numCourses;i++)

    if(courses[i]==c){

        int k=0;

    String temp[] = new String[numCourses-1];

    for(i=0;i<numCourses;i++)

    if(courses[i]!=c)

        temp[k++] = courses[i];

        return true;

    }

    return false;

}

    String tostring(){

        return "Teacher: "+super.tostring();

    }

}

class personDriver{

public static void main(String[] args){

    studdent stud = new studdent("jyotikrish","silicon");

    String c[] = {"course A","course B","course C"};

    int grad[] = {40,78,97};

    stud.addCourseGrade(c,grad);

    stud.printGrades();

    System.out.println("Average : "+stud.getAverageGrade()+"\n"+stud.tostring());

    Teacher th = new Teacher("theacher one", "bbsr2");

    System.out.println(th.addCourse("course-1"));

    System.out.println(th.addCourse("course-2"));

    System.out.println(th.addCourse("course-1"));

    System.out.println(th.addCourse("course-3"));

    System.out.println(th.removeCourse("course-3"));

    System.out.println(th.removeCourse("course-5"));

    System.out.println(th.tostring());

    }

}

2.

class Shape{

    String color;

    boolean filled;

    Shape(){

        setCOLOR("green");

        setFILLED(true);

    }

    Shape(String color1,boolean filled1){

        setCOLOR(color1);

        setFILLED(filled1);

    }

    String getCOLOR(){

        return color;

    }

    void setCOLOR(String color1){

        color=color1;

    }

    boolean getFILLED(){

        return filled;

    }

    void setFILLED(boolean filled1){

        filled=filled1;

    }

    String tostring(){

        return "A shape with color of"+getCOLOR()+" and "+getFILLED();

    }

}

class Circle extends Shape{

    double radius;

    Circle(){setRADIUS(1.0);}

    Circle(double radius1){

        setRADIUS(radius1);

    }

    double getRADIUS(){

        return radius;

    }

    void setRADIUS(double radius1){

        radius=radius1;

    }

    double getArea(){

        return 3.14\*radius\*radius;

    }

    double getPerimeter(){

        return 2\*3.14\*radius;

    }

    String tostring(){

        return "A Circle with radius ="+getRADIUS()+" which is a subclass of "+super.tostring();

    }

}

class Rectangle extends Shape{

    double width,length;

    Rectangle(){

        setWIDTH(1.0);

        setLENGTH(1.0);

    }

    Rectangle(double width1,double length1){

        setWIDTH(width1);

        setLENGTH(length1);

    }

    double getWIDTH(){

        return width;

    }

    double getLENGTH(){

        return length;

    }

    void setWIDTH(double width1){

        width=width1;

    }

    void setLENGTH(double length1){

        length=length1;

    }

    double getArea(){

        return width\*length;

    }

    double getPerimeter(){

        return 2\*(width+length);

    }

    String tostring(){

        return "A Rectangle with width ="+getWIDTH()+" and length"+getLENGTH()+" which is a subclass of "+super.tostring();

    }

}

class Square extends Rectangle{

    Square(){}

    Square(double side){

        super(side,side);

    }

    String tostring(){

        return "A Square with side ="+getWIDTH()+" which is a subclass of "+super.tostring();

    }

    void setLENGTH(double l)

    {

        super.setLENGTH(l);

    }

    void setWIDTH(double w)

    {

        super.setWIDTH(w);

    }

}

class ptDriver

{

    public static void main(String[] args) {

        Shape s = new Shape();

        Circle p1 = new Circle(50);

        System.out.println(p1.tostring());

        Rectangle r1 =new Rectangle(2.0,4.0);

        System.out.println(r1.tostring());

        Square s1= new Square(2.0);

        System.out.println(s1.tostring());

    }

}

3.

*//i changed the variable name (not as in question) as they were conflicting with previous program...*

abstract class absShape{

    protected String color;

    protected boolean filled;

    absShape(){

        setCOLOR("green");

        setFILLED(true);

    }

    absShape(String color1,boolean filled1){

        setCOLOR(color1);

        setFILLED(filled1);

    }

    String getCOLOR(){

        return color;

    }

    void setCOLOR(String color1){

        color=color1;

    }

    boolean isFILLED(){

        return filled;

    }

    void setFILLED(boolean filled1){

        filled=filled1;

    }

    abstract String tostring();

*//{*

*//  return "A shape with color of"+getCOLOR()+" and "+isFILLED();*

*//}*

    abstract double getArea();

    abstract double getPerimeter();

}

class Circle extends absShape{

    protected double radius;

    Circle(){setRADIUS(0.0);}

    Circle(double radius1){

        setRADIUS(radius1);

    }

    Circle(double radius1,String color1,boolean filled1){

        setRADIUS(radius1);

        setCOLOR(color1);

        setFILLED(filled1);

    }

    double getRADIUS(){

        return radius;

    }

    void setRADIUS(double radius1){

        radius=radius1;

    }

    double getArea(){

        return 3.14\*radius\*radius;

    }

    double getPerimeter(){

        return 2\*3.14\*radius;

    }

    String tostring(){

        return "A Circle with radius ="+getRADIUS()+" which is a subclass of "+tostring();

    }

}

class Rectangle extends absShape{

    protected double width,length;

    Rectangle(){

        setWIDTH(0.0);

        setLENGTH(0.0);

    }

    Rectangle(double width1,double length1){

        setWIDTH(width1);

        setLENGTH(length1);

    }

    Rectangle(double width1,double length1,String color1,boolean filled1){

        setWIDTH(width1);

        setLENGTH(length1);

        setCOLOR(color1);

        setFILLED(filled1);

    }

    double getWIDTH(){

        return width;

    }

    double getLENGTH(){

        return length;

    }

    void setWIDTH(double width1){

        width=width1;

    }

    void setLENGTH(double length1){

        length=length1;

    }

    double getArea(){

        return width\*length;

    }

    double getPerimeter(){

        return 2\*(width+length);

    }

    String tostring(){

        return "A Rectangle with width ="+getWIDTH()+" and length"+getLENGTH()+" which is a subclass of "+tostring();

    }

}

class Square extends Rectangle{

    double side;

    Square(){}

    Square(double side){

        setSIDE(side);

    }

    Square(double side1,String color1,boolean filled1){

        setSIDE(side1);

        setCOLOR(color1);

        setFILLED(filled1);

    }

    double getSIDE(){

        return side;

    }

    void setSIDE(double side1){

        side=side1;

    }

    String tostring(){

        return "A Square with side ="+getSIDE()+" which is a subclass of "+tostring();

    }

    void setLENGTH(double side)

    {

        super.setLENGTH(side);

    }

    void setWIDTH(double side)

    {

        super.setWIDTH(side);

    }

}

class ptDriver1

{

    public static void main(String[] args) {

        absShape s1 = new Circle(5.5, "RED", false);

        System.out.println(s1);

        System.out.println(s1.getArea());

        System.out.println(s1.getPerimeter());

        System.out.println(s1.getCOLOR());

        System.out.println(s1.isFILLED());

*//System.out.println(s1.getRADIUS());//here error occurs as it protected*

        Circle c1 = (Circle)s1;

        System.out.println(c1);

        System.out.println(c1.getArea());

        System.out.println(c1.getPerimeter());

        System.out.println(c1.getCOLOR());

        System.out.println(c1.isFILLED());

        System.out.println(c1.getRADIUS());

*//absShape s2 = new absShape();//abstract class cant be instantiated*

        absShape s3 = new Rectangle(1.0, 2.0, "RED", false);

        System.out.println(s3);

        System.out.println(s3.getArea());

        System.out.println(s3.getPerimeter());

        System.out.println(s3.getCOLOR());

*//System.out.println(s3.getLENGTH());*

        Rectangle r1 = (Rectangle)s3; *// downcast*

        System.out.println(r1);

        System.out.println(r1.getArea());

        System.out.println(r1.getCOLOR());

        System.out.println(r1.getLENGTH());

        absShape s4 = new Square(6.6); *// Upcast*

        System.out.println(s4);

        System.out.println(s4.getArea());

        System.out.println(s4.getCOLOR());

*//System.out.println(s4.getSIDE());*

        Rectangle r2 = (Rectangle)s4;

        System.out.println(r2);

        System.out.println(r2.getArea());

        System.out.println(r2.getCOLOR());

*//System.out.println(r2.getSIDE());*

        System.out.println(r2.getLENGTH());

    }

}