UCS1313 Object Oriented Programming using Java Lab
AY: 2021-22

Name: Krithika Swaminathan
Roll No.: 205001057

Assignment 4 – Abstract Class and Interfaces

Q1: Design a class called Person as described. A sub-class Employee of class Person is designed as shown. A sub-class Faculty of class Employee is designed as shown. Design an Interface Student. Design a sub-class TeachingAssistant of class Employee, implements <<Student>>. Write a TestDriver function to get input for Faculty and TeachingAssistant and display their details. Find the class that can be kept as abstract.

Code:

```
import java.util.Scanner;

abstract class Person {
        //data members
        private String name;
        private String address;
        //constructor
        Person (String aName, String anAddress) {
                name = aName;
                address = anAddress;
                }
        //member functions
        String getName() {
                return name;
                }
        String getAddress() {
                return address;
                }
        void setAddress(String anAddress) {
                address = anAddress;
                }
        abstract double calSalary();
        }

class Employee extends Person {
        //data members
        private String empid;
        private String dept;
        private int basic;
        //constructor
        Employee (String aName, String anAddress, String anEmpid, String aDept, int aBasic) {
                super(aName,anAddress);
                empid = anEmpid;
                dept = aDept;
                basic = aBasic;
                }
        //member functions
        int getEmpid() {
```

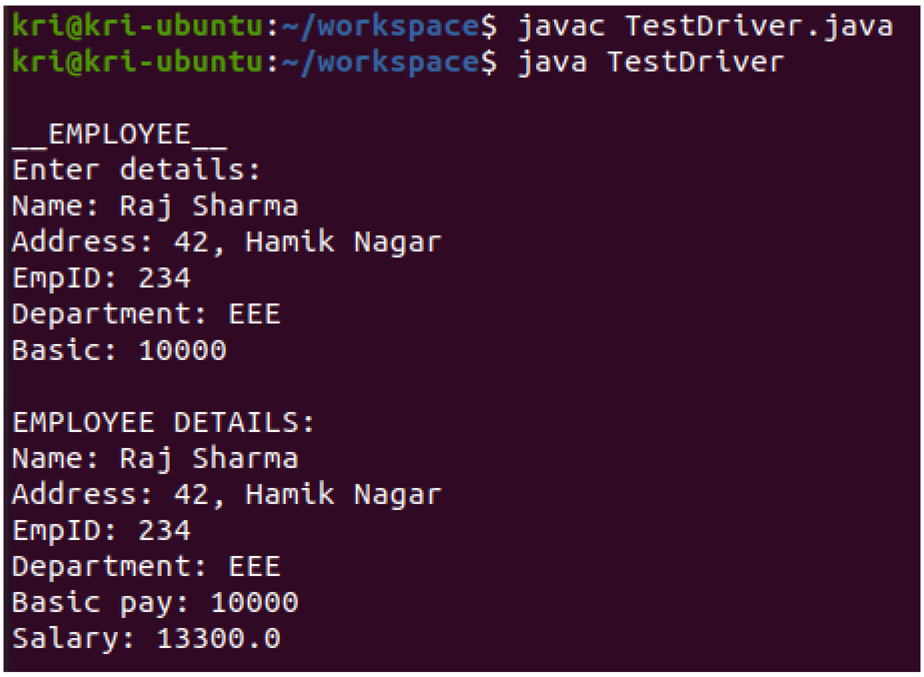
```
		return Integer.parseInt(empid);
		}
	String getDept() {
		return dept;
		}
	int getBasic() {
		return basic;
		}
	void setDept (String aDept) {
		dept = aDept;
		}
	void setBasic (int aBasic) {
		basic = aBasic;
		}
	double calSalary() {
		int bas = getBasic();
		double DA = 0.4*bas, HRA = 0.1*bas, MedIns = 0.09*bas, PF = 0.08*bas;
		double GS = bas + DA + HRA;
		double Deduction = MedIns + PF;
		return GS - Deduction;
		}

	void display() {
		System.out.println("\nEMPLOYEE DETAILS:");
		System.out.println("Name: "+getName());
		System.out.println("Address: "+getAddress());
		System.out.println("EmpID: "+getEmpid());
		System.out.println("Department: "+getDept());
		System.out.println("Basic pay: "+getBasic());
		System.out.println("Salary: "+calSalary());
		}
	}

class Faculty extends Employee {
	//data members
	private String designation;
	private String course;
	//constructor
	Faculty (String aName, String anAddress, String anEmpid, String aDept, int aBasic, String de,
String aCourse) {
		super(aName,anAddress,anEmpid,aDept,aBasic);
		designation = de;
		course = aCourse;
		}
	//member functions
	String getDesig() {
		return designation;
		}
	float getCourse() {
		return Float.parseFloat(course);
		}
	void setDesig (String aDept) {
		designation = aDept;
		}
	void setCourse (String aCourse) {
		course = aCourse;
		}
	double calSalary() {
		int bas = getBasic();
		double DA = 0.4*bas, HRA = 0.1*bas, MedIns = 0.09*bas, PF = 0.08*bas;
		double GS = bas + DA + HRA;
		double Deduction = MedIns + PF;
		return GS - Deduction;
		}

	void display() {
		System.out.println("\nFACULTY DETAILS:");
		System.out.println("Name: "+getName());
		System.out.println("Address: "+getAddress());
		System.out.println("EmpID: "+getEmpid());
		System.out.println("Department: "+getDept());
		System.out.println("Basic pay: "+getBasic());
		System.out.println("Designation: "+designation);
		System.out.println("Course: "+getCourse());
		System.out.println("Salary: "+calSalary());
		}
	}

interface Student {
	float [] getMarks();
	float calcGPA();
	}

class TeachingAssistant extends Employee implements Student {
	//data members
	private String project;
	private String course;
	private float marks[];
	//constructor
	TeachingAssistant (String aName, String anAddress, String anEmpid, String aDept, int aBasic,
String aProject, String aCourse, float arrMarks[]) {
		super(aName,anAddress,anEmpid,aDept,aBasic);
		project = aProject;
		course = aCourse;
		marks = new float[3];
		for (int i=0; i<arrMarks.length; i++) {
			marks[i] = arrMarks[i];
			}
			}
		//member functions
		String getProject() {
			return project;
			}
		String getCourse() {
			return course;
			}
		public float[] getMarks() { //check input-ouput type for func
			return marks;
			}
		void setCourse (String aCourse) {
			course = aCourse;
			}
		public float calcGPA() {
			float GPA = 0;
			for (int i=0; i<3; i++)
				GPA += (int)(marks[i]/10);
			GPA/=3;
			return GPA;
			}
		double calSalary() {
			int bas = getBasic();
			double DA = 0.4*bas, HRA = 0.1*bas, MedIns = 0.09*bas, PF = 0.08*bas;
			double GS = bas + DA + HRA;
			double Deduction = MedIns + PF;
			return GS - Deduction;
			}

		void display() {
			System.out.println("\nTEACHING ASSISTANT DETAILS:");
			System.out.println("Name: "+getName());
			System.out.println("Address: "+getAddress());
			System.out.println("EmpID: "+getEmpid());
			System.out.println("Department: "+getDept());
			System.out.println("Basic pay: "+getBasic());
			System.out.println("Project: "+getProject());
			System.out.println("Course: "+getCourse());
			System.out.print("Marks: ");
			for (int i=0; i<marks.length; i++)
				System.out.print(marks[i]+"  ");
			System.out.println("\nGPA: "+calcGPA());
			System.out.println("Salary: "+calSalary());
			}
		}

class TestDriver {
		public static void main (String arg[]) {
			//declaring new scanner object for input
```

```
Scanner sc = new Scanner(System.in);

//getting data from the user

//class 1 - person
String name, address;

//class 2 - employee
String empid, dept;
int basic;
System.out.println("\n__EMPLOYEE__\nEnter details: ");
System.out.print("Name: ");
name = sc.nextLine();
System.out.print("Address: ");
address = sc.nextLine();
System.out.print("EmpID: ");
empid = sc.next();
System.out.print("Department: ");
dept = sc.next();
System.out.print("Basic: ");
basic = sc.nextInt();
sc.nextLine();

Employee E = new Employee(name,address,empid,dept,basic);
E.display();

//class 3 - faculty
String desgn, course;
System.out.println("\n__FACULTY__\nEnter details: ");
System.out.print("Name: ");
name = sc.nextLine();
System.out.print("Address: ");
address = sc.nextLine();
System.out.print("EmpID: ");
empid = sc.next();
System.out.print("Department: ");
dept = sc.next();
System.out.print("Basic: ");
basic = sc.nextInt();
sc.nextLine();
System.out.print("Designation: ");
desgn = sc.nextLine();
System.out.print("Course: ");
course = sc.nextLine();

Faculty F = new Faculty(name,address,empid,dept,basic,desgn,course);
F.display();

//class 4 - teaching assistant
```

Name: Krithika Swaminathan
Roll No.: 205001057

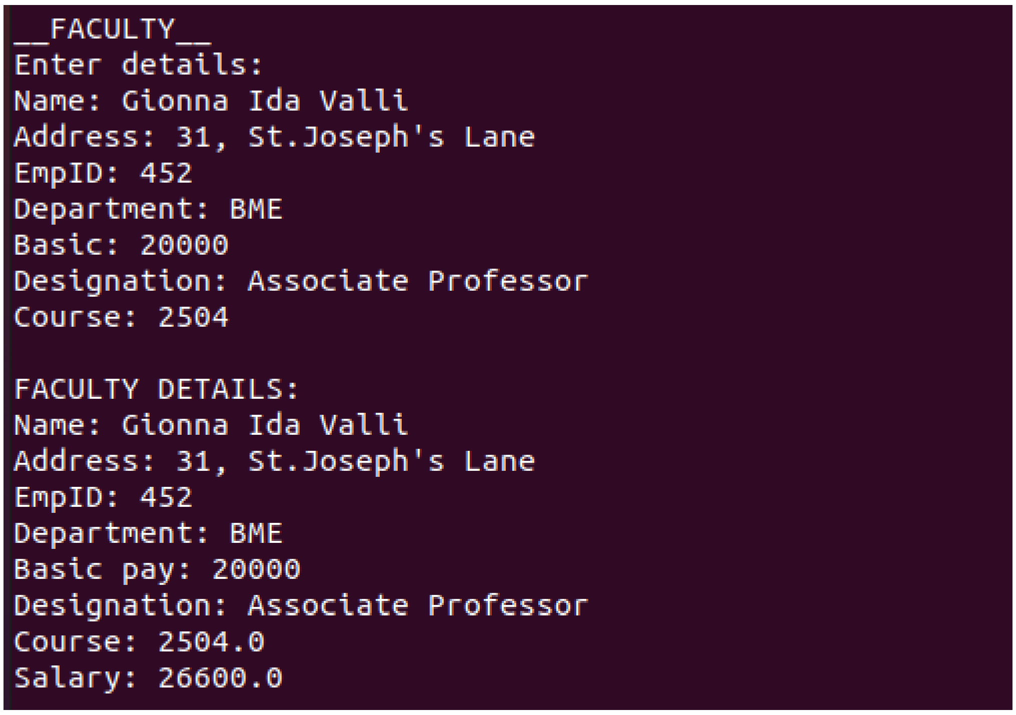
```
			String project;
			float [] marks = new float[3];
			System.out.println("\n__TEACHING ASSISTANT__\nEnter details: ");
			System.out.print("Name: ");
			name = sc.nextLine();
			System.out.print("Address: ");
			address = sc.nextLine();
			System.out.print("EmpID: ");
			empid = sc.next();
			System.out.print("Department: ");
			dept = sc.next();
			System.out.print("Basic: ");
			basic = sc.nextInt();
			sc.nextLine();
			System.out.print("Designation: ");
			desgn = sc.nextLine();
			System.out.print("Course: ");
			course = sc.nextLine();
			System.out.print("Project: ");
			project = sc.nextLine();
			for (int i=0; i<3; i++) {
				System.out.print("Mark"+(i+1)+": ");
				marks[i] = sc.nextFloat();
				}

			TeachingAssistant TA = new
TeachingAssistant(name,address,empid,dept,basic,project,course,marks);
			TA.display();

			}
		}
```

Output:

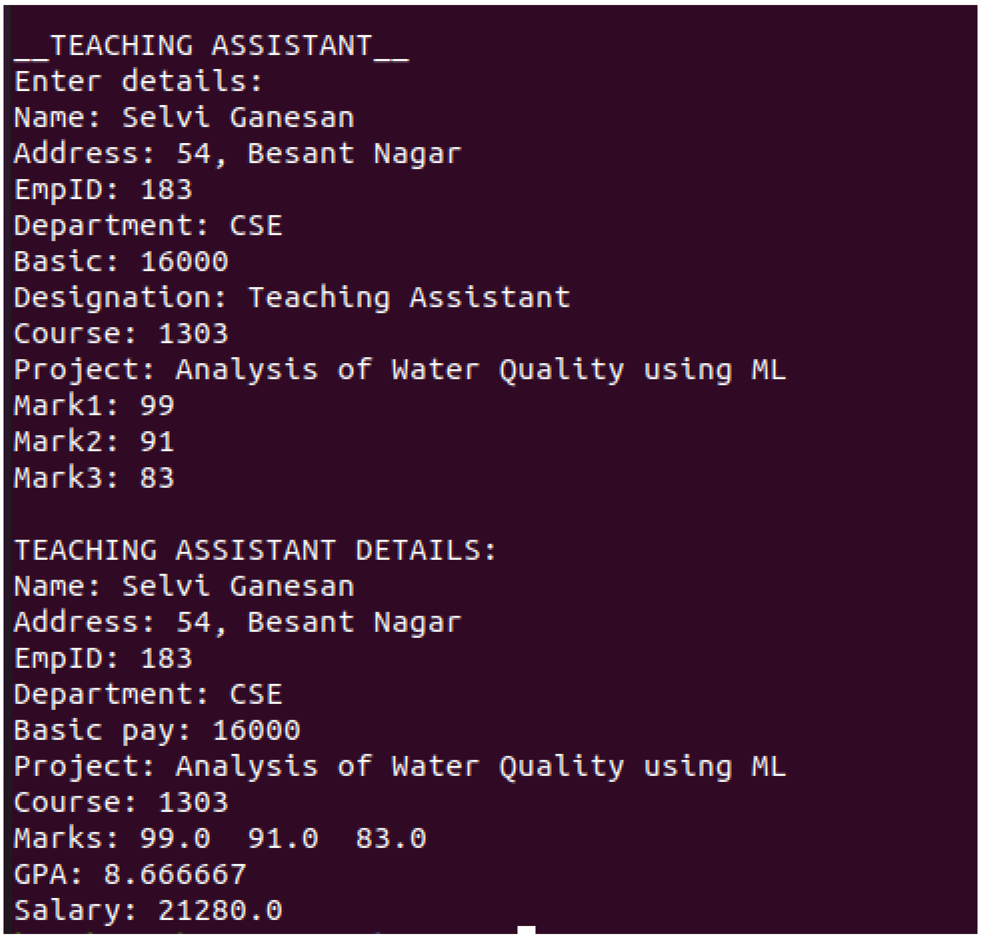


```
kri@kri-ubuntu:~/workspace$ javac TestDriver.java
kri@kri-ubuntu:~/workspace$ java TestDriver

__EMPLOYEE__
Enter details:
Name: Raj Sharma
Address: 42, Hamik Nagar
EmpID: 234
Department: EEE
Basic: 10000

EMPLOYEE DETAILS:
Name: Raj Sharma
Address: 42, Hamik Nagar
EmpID: 234
Department: EEE
Basic pay: 10000
Salary: 13300.0
```

Name: Krithika Swaminathan
Roll No.: 205001057



```
__FACULTY__
Enter details:
Name: Gionna Ida Valli
Address: 31, St.Joseph's Lane
EmpID: 452
Department: BME
Basic: 20000
Designation: Associate Professor
Course: 2504

FACULTY DETAILS:
Name: Gionna Ida Valli
Address: 31, St.Joseph's Lane
EmpID: 452
Department: BME
Basic pay: 20000
Designation: Associate Professor
Course: 2504.0
Salary: 26600.0
```



```
__TEACHING ASSISTANT__
Enter details:
Name: Selvi Ganesan
Address: 54, Besant Nagar
EmpID: 183
Department: CSE
Basic: 16000
Designation: Teaching Assistant
Course: 1303
Project: Analysis of Water Quality using ML
Mark1: 99
Mark2: 91
Mark3: 83

TEACHING ASSISTANT DETAILS:
Name: Selvi Ganesan
Address: 54, Besant Nagar
EmpID: 183
Department: CSE
Basic pay: 16000
Project: Analysis of Water Quality using ML
Course: 1303
Marks: 99.0  91.0  83.0
GPA: 8.666667
Salary: 21280.0
```

Name: Krithika Swaminathan
Roll No.: 205001057

Q2: Create a class hierarchy for the following using Interface / Abstract class. Design Shape as described. A sub-class Circle of class Shape is designed as shown. A sub-class Rectangle of class Shape is designed as shown. A sub-class Square of class rectangle designed as shown.

Code:

```
import java.util.Scanner;

abstract class Shape {
	//data members
	protected String color;

	//constructors
	Shape() { color = "red"; }
	Shape(String col) { color = col; }

	//public methods
	String getColor() { return color; }
	void setColor(String col) { color = col; }

	//declaring abstract methods
	abstract double getArea();
	abstract double getPerimeter();

	//declaring dummy methods to facilitate method overriding
	float getRadius() { return 0; }
	void setRadius(float none) {}
	float getWidth() {return 0; }
	void setWidth(float none) {}
	float getLength() { return 0; }
	void setLength(float none) {}
	float getSide() { return 0; }
	void setSide(float none) {}
	}

class Circle extends Shape {
	//data members
	protected float radius;
	//constructors
	Circle() {
		super();
		radius = 1;
		}
	Circle(float r) {
		super();
		radius = r;
		}
	Circle(float r, String col) {
			super(col);
			radius = r;
			}
		//public methods
		float getRadius() { return radius; }
		void setRadius(float r) { radius = r; }
		double getArea() {
			return 3.14*radius*radius;
			}
		double getPerimeter() {
			return 2*3.14*radius;
			}
		}

class Rectangle extends Shape {
		//data members
		protected float width;
		protected float length;
		//constructors
		Rectangle() {
			super();
			width = 1;
			length = 1;
			}
		Rectangle(float w, float l) {
			super();
			width = w;
			length = l;
			}
		Rectangle(float w, float l, String col) {
			super(col);
			width = w;
			length = l;
			}
		//public methods
		float getWidth() { return width; }
		void setWidth(float w) { width = w; }
		float getLength() { return length; }
		void setLength(float l) { length = l; }
		double getArea() {
			return length*width;
			}
		double getPerimeter() {
			return 2*(length+width);
			}
		}

class Square extends Rectangle {
		//constructors
```

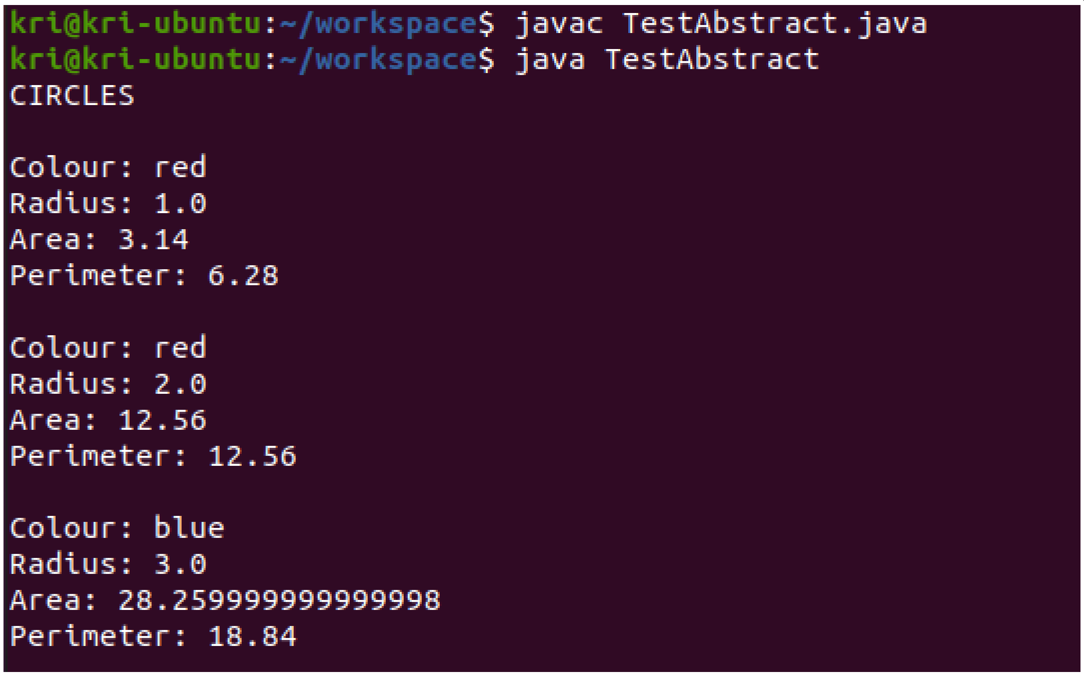
```
	Square() {
		super();
		}
	Square(float side) {
		super(side,side);
		}
	Square(float side, String col) {
		super(side,side,col);
		}
	//public methods
	float getSide() { return length; }
	void setSide(float side) {
		length = side;
		width = side;
		}
	}

class TestAbstract {
	public static void main (String a[]) {
		//declaring new scanner object
		Scanner sc = new Scanner(System.in);

		//declaring a 2D array of shapes - each row contains a different shape
		Shape shapes[][] = new Shape[3][3];

		shapes[0][0] = new Circle();
		shapes[0][1] = new Circle(2);
		shapes[0][2] = new Circle(3,"blue");

		shapes[1][0] = new Rectangle();
		shapes[1][1] = new Rectangle(5,7);
		shapes[1][2] = new Rectangle(8,3,"yellow");

		shapes[2][0] = new Square();
		shapes[2][1] = new Square(4);
		shapes[2][2] = new Square(2,"purple");

		//displaying values
		for (int i=0; i<shapes.length; i++) {
			if (i==0) System.out.println("CIRCLES\n");
			else if (i==1) System.out.println("RECTANGLES\n");
			else System.out.println("SQUARES\n");

			for (int j=0; j<shapes.length; j++) {
				System.out.println("Colour: "+shapes[i][j].getColor());
				switch(i) {
					case 0: {
						System.out.println("Radius: "+shapes[i][j].getRadius());
						break;
```

```
                        }
                    case 1: {
                        System.out.println("Width: "+shapes[i][j].getWidth());
                        System.out.println("Length: "+shapes[i][j].getLength());
                        break;
                        }
                    case 2: {
                        System.out.println("Side: "+shapes[i][j].getSide());
                        break;
                        }
                    default: System.exit(0);
                    }
                System.out.println("Area: "+shapes[i][j].getArea());
                System.out.println("Perimeter: "+shapes[i][j].getPerimeter()+'\n');
                }
        }

    //getting new values from the user
    System.out.println("\nEnter new values: ");
    System.out.print("Enter new color: ");
    String col = sc.next();
    System.out.print("Enter new radius: ");
    float rad = sc.nextFloat();
    System.out.print("Enter new width: ");
    float wid = sc.nextFloat();
    System.out.print("Enter new length: ");
    float len = sc.nextFloat();

    //setting new values for objects
    System.out.println("\nNew values:");
    for (int i=0; i<shapes.length; i++) {
        if (i==0) System.out.println("CIRCLES\n");
        else if (i==1) System.out.println("RECTANGLES\n");
        else System.out.println("SQUARES\n");

        for (int j=0; j<shapes.length; j++) {
            shapes[i][j].setColor(col);
            System.out.println("Colour: "+shapes[i][j].getColor());
            switch(i) {
                case 0: {
                    shapes[i][j].setRadius(rad);
                    System.out.println("Radius: "+shapes[i][j].getRadius());
                    break;
                    }
                case 1: {
                    shapes[i][j].setWidth(wid);
                    shapes[i][j].setLength(len);
                    System.out.println("Width: "+shapes[i][j].getWidth());
                    System.out.println("Length: "+shapes[i][j].getLength());
```

```
                                                break;
                                                }
                                        case 2: {
                                                if (len != wid) {
                                                        System.out.println("Length and width are different -->
Square not possible");

                                                        System.out.println("Considering 'length' to be the side

of the square...");

                                                        }
                                                shapes[i][j].setSide(len);
                                                System.out.println("Side: "+shapes[i][j].getSide());
                                                break;
                                                }
                                        default: System.exit(0);
                                        }
                                System.out.println("Area: "+shapes[i][j].getArea());
                                System.out.println("Perimeter: "+shapes[i][j].getPerimeter()+'\n');
                                }
                        }
                }
        }
```

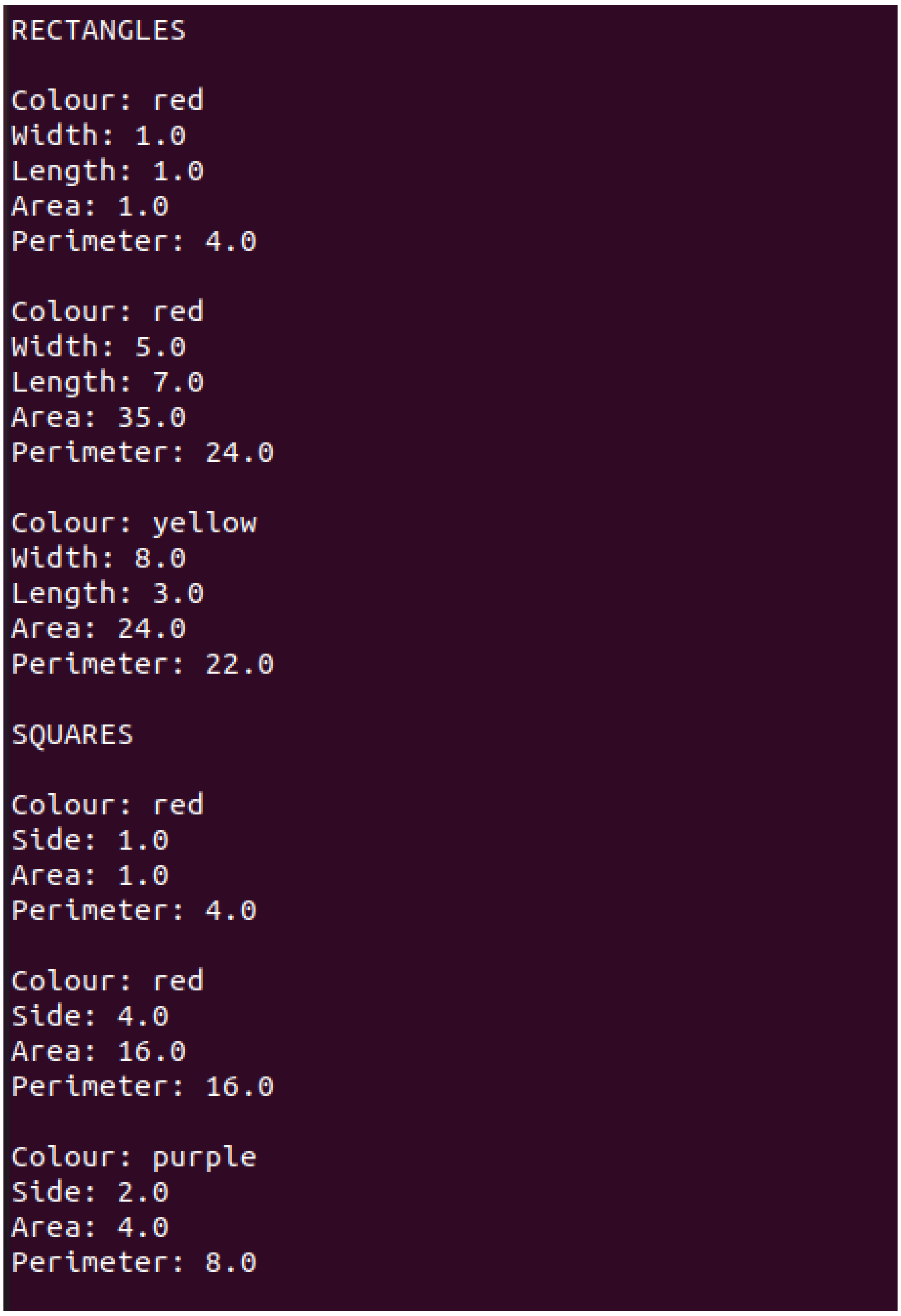
Output:

(Test case 1)

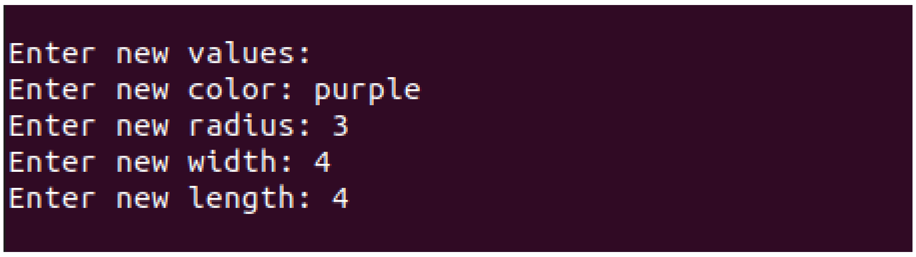


```
kri@kri-ubuntu:~/workspace$ javac TestAbstract.java
kri@kri-ubuntu:~/workspace$ java TestAbstract
CIRCLES

Colour: red
Radius: 1.0
Area: 3.14
Perimeter: 6.28

Colour: red
Radius: 2.0
Area: 12.56
Perimeter: 12.56

Colour: blue
Radius: 3.0
Area: 28.259999999999998
Perimeter: 18.84
```

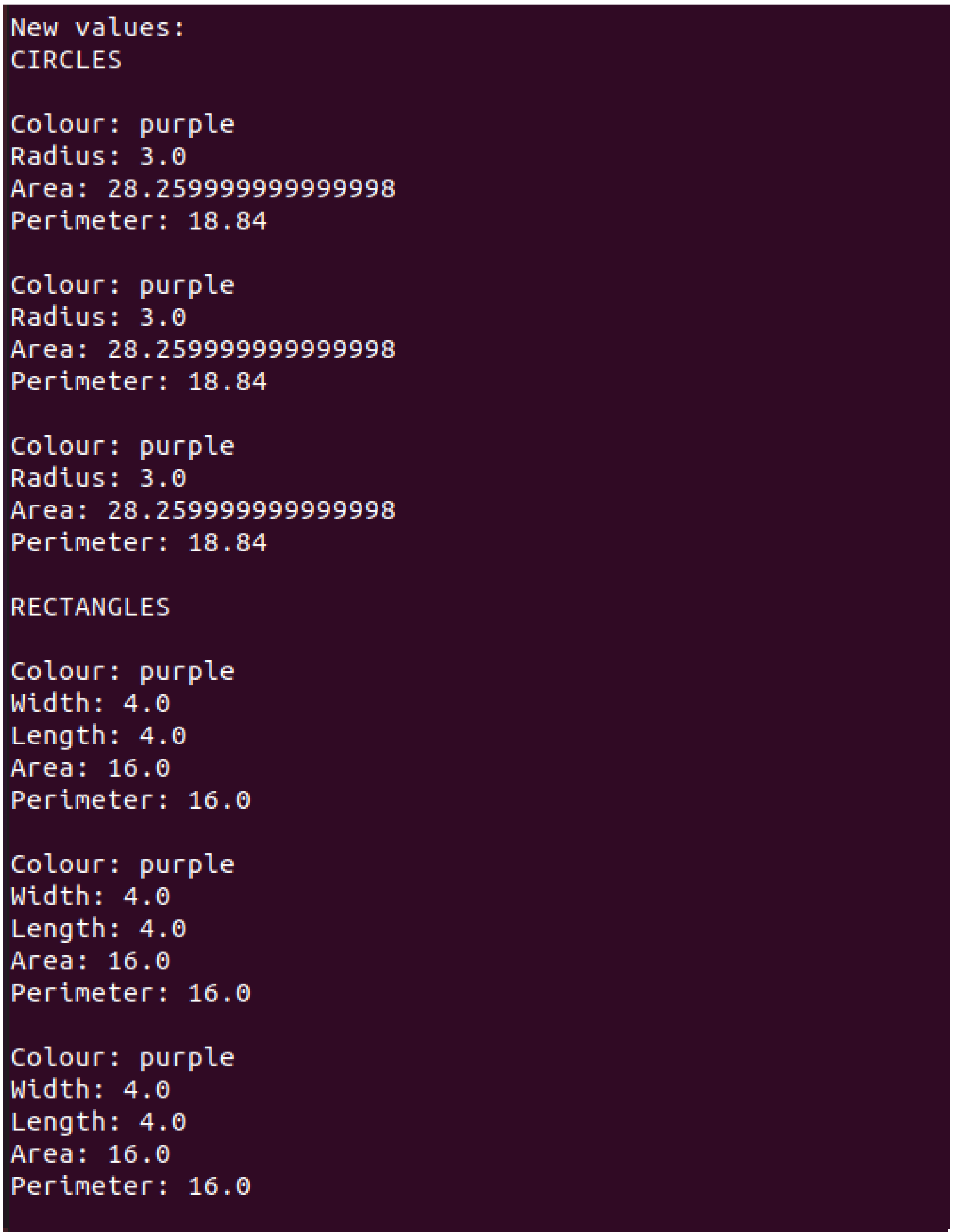
Name: Krithika Swaminathan
Roll No.: 205001057



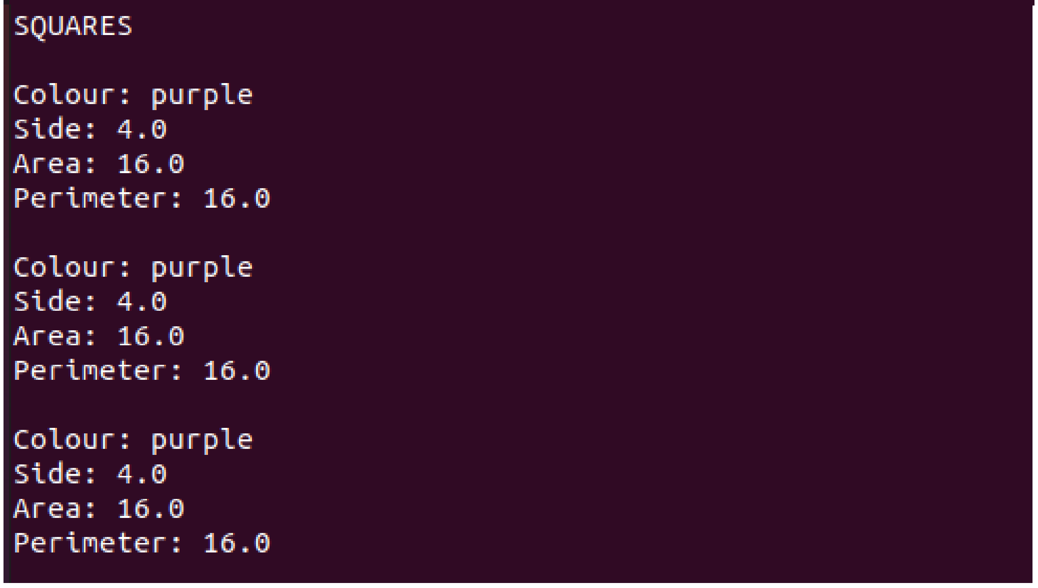
```
RECTANGLES

Colour: red
Width: 1.0
Length: 1.0
Area: 1.0
Perimeter: 4.0

Colour: red
Width: 5.0
Length: 7.0
Area: 35.0
Perimeter: 24.0

Colour: yellow
Width: 8.0
Length: 3.0
Area: 24.0
Perimeter: 22.0

SQUARES

Colour: red
Side: 1.0
Area: 1.0
Perimeter: 4.0

Colour: red
Side: 4.0
Area: 16.0
Perimeter: 16.0

Colour: purple
Side: 2.0
Area: 4.0
Perimeter: 8.0
```



```
Enter new values:
Enter new color: purple
Enter new radius: 3
Enter new width: 4
Enter new length: 4
```



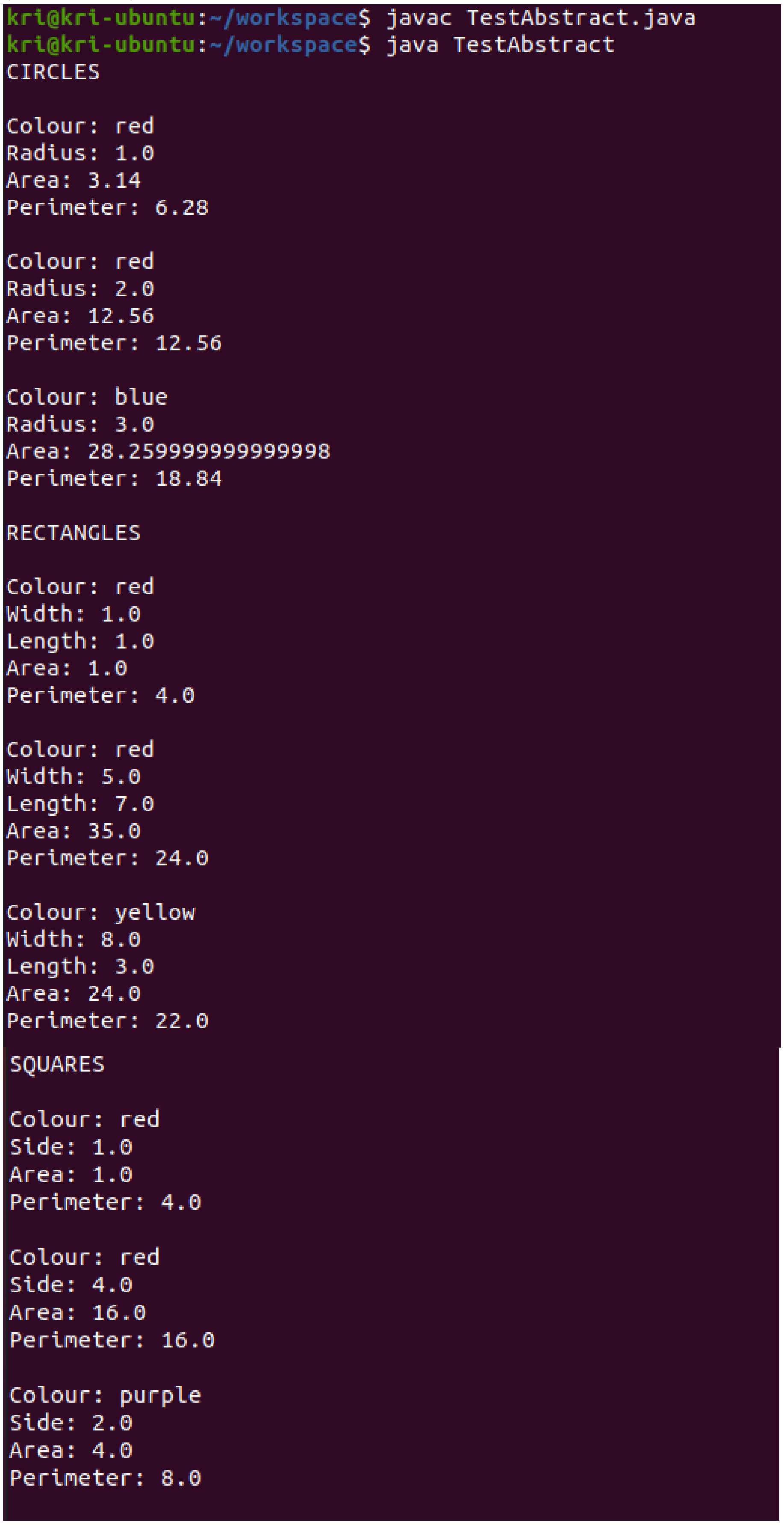
```
New values:
CIRCLES

Colour: purple
Radius: 3.0
Area: 28.259999999999998
Perimeter: 18.84

Colour: purple
Radius: 3.0
Area: 28.259999999999998
Perimeter: 18.84

Colour: purple
Radius: 3.0
Area: 28.259999999999998
Perimeter: 18.84

RECTANGLES

Colour: purple
Width: 4.0
Length: 4.0
Area: 16.0
Perimeter: 16.0

Colour: purple
Width: 4.0
Length: 4.0
Area: 16.0
Perimeter: 16.0

Colour: purple
Width: 4.0
Length: 4.0
Area: 16.0
Perimeter: 16.0
```



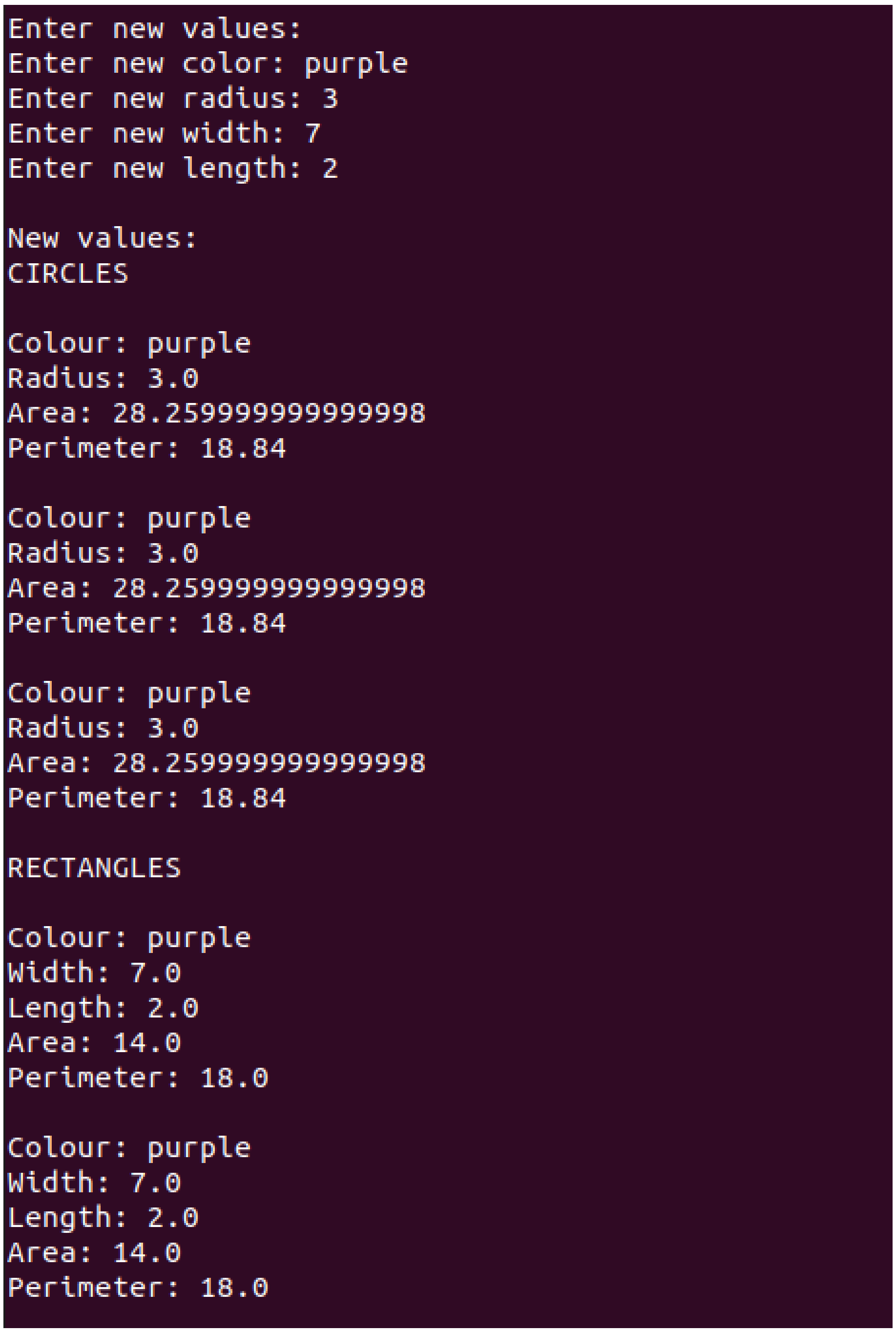
```
SQUARES

Colour: purple
Side: 4.0
Area: 16.0
Perimeter: 16.0

Colour: purple
Side: 4.0
Area: 16.0
Perimeter: 16.0

Colour: purple
Side: 4.0
Area: 16.0
Perimeter: 16.0
```

Name: *Krithika Swaminathan*
Roll No.: *205001057*

(Test case 2)

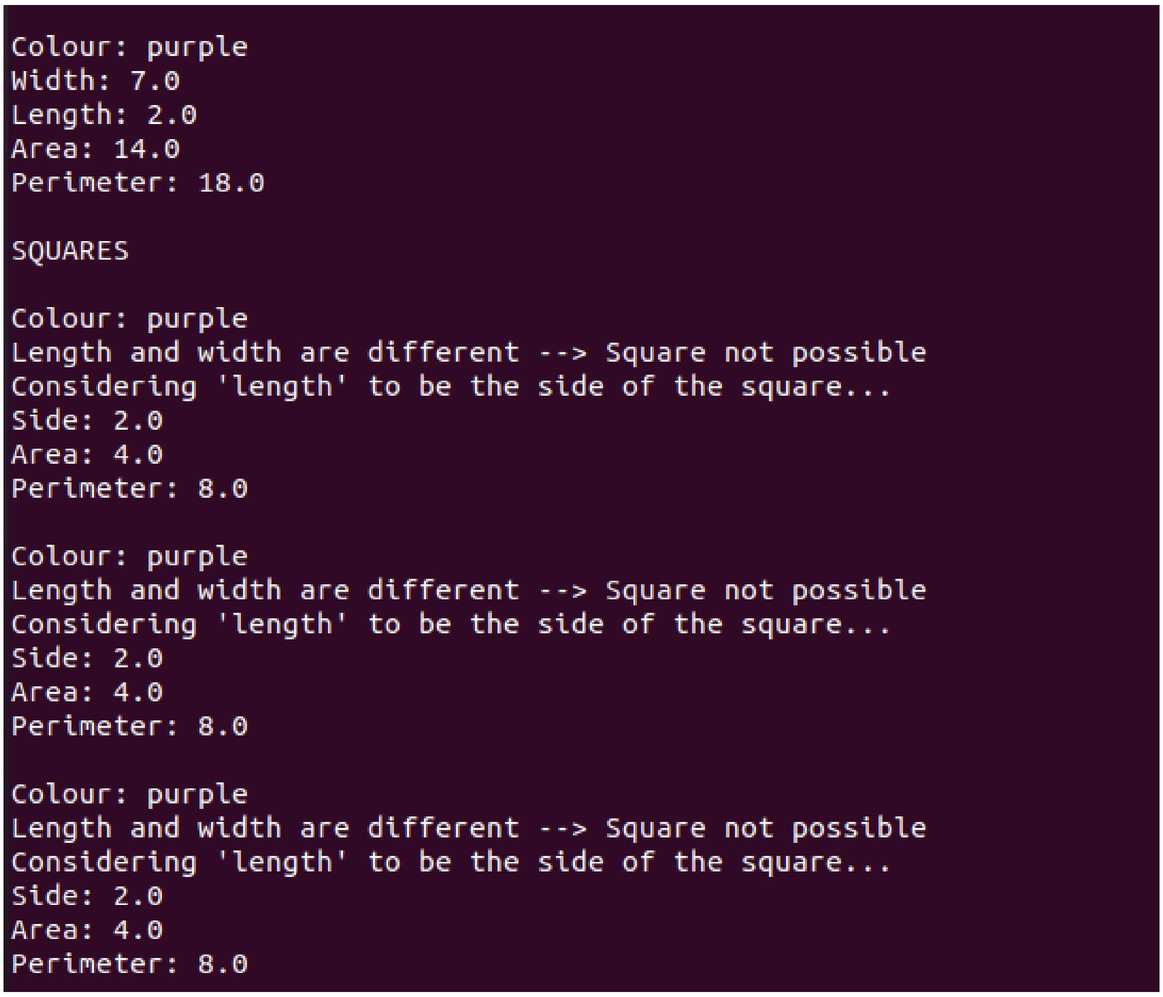


```
kri@kri-ubuntu:~/workspace$ javac TestAbstract.java
kri@kri-ubuntu:~/workspace$ java TestAbstract
CIRCLES

Colour: red
Radius: 1.0
Area: 3.14
Perimeter: 6.28

Colour: red
Radius: 2.0
Area: 12.56
Perimeter: 12.56

Colour: blue
Radius: 3.0
Area: 28.259999999999998
Perimeter: 18.84

RECTANGLES

Colour: red
Width: 1.0
Length: 1.0
Area: 1.0
Perimeter: 4.0

Colour: red
Width: 5.0
Length: 7.0
Area: 35.0
Perimeter: 24.0

Colour: yellow
Width: 8.0
Length: 3.0
Area: 24.0
Perimeter: 22.0
SQUARES

Colour: red
Side: 1.0
Area: 1.0
Perimeter: 4.0

Colour: red
Side: 4.0
Area: 16.0
Perimeter: 16.0

Colour: purple
Side: 2.0
Area: 4.0
Perimeter: 8.0
```



```
Enter new values:
Enter new color: purple
Enter new radius: 3
Enter new width: 7
Enter new length: 2

New values:
CIRCLES

Colour: purple
Radius: 3.0
Area: 28.259999999999998
Perimeter: 18.84

Colour: purple
Radius: 3.0
Area: 28.259999999999998
Perimeter: 18.84

Colour: purple
Radius: 3.0
Area: 28.259999999999998
Perimeter: 18.84

RECTANGLES

Colour: purple
Width: 7.0
Length: 2.0
Area: 14.0
Perimeter: 18.0

Colour: purple
Width: 7.0
Length: 2.0
Area: 14.0
Perimeter: 18.0
```

Name: Krithika Swaminathan
Roll No.: 205001057



```
Colour: purple
Width: 7.0
Length: 2.0
Area: 14.0
Perimeter: 18.0

SQUARES

Colour: purple
Length and width are different --> Square not possible
Considering 'length' to be the side of the square...
Side: 2.0
Area: 4.0
Perimeter: 8.0

Colour: purple
Length and width are different --> Square not possible
Considering 'length' to be the side of the square...
Side: 2.0
Area: 4.0
Perimeter: 8.0

Colour: purple
Length and width are different --> Square not possible
Considering 'length' to be the side of the square...
Side: 2.0
Area: 4.0
Perimeter: 8.0
```