

JAVA PROGRAMMING ASSIGNMENT 4

Name:PUNEETH L

USN: 1BM24MC069

1. Provided that you have a given number of small rice bags (1 kilo each) and big rice bags (5 kilos each), write a method that returns true if it is possible to make a package with goal kilos of rice.

PROGRAM:

```
import java.util.*;

class Ricebags {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of small rice bags (1 kilo each): ");

        int small = sc.nextInt();

        System.out.print("Enter the number of big rice bags (5 kilo each): ");

        int big = sc.nextInt();

        System.out.print("Enter the kilos of rice: ");

        int r_kilo = sc.nextInt();

        boolean possible = isPossible(small, big, r_kilo);

        if (possible) {

            System.out.println("It is possible to pack the rice.");

        } else {

            System.out.println("It is NOT possible to pack the rice.");

        }

    }

    static boolean isPossible(int small, int big, int r_kilo) {

        int maxBigBags = r_kilo / 5;

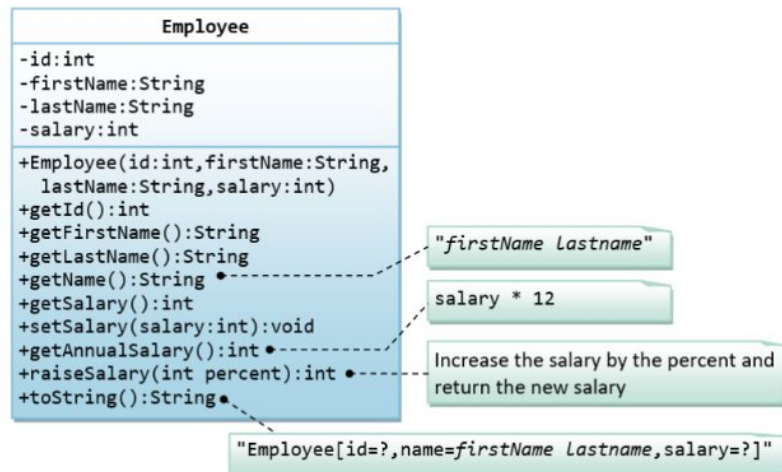
        int useBig = Math.min(big, maxBigBags);
```

```
    int remaining = r_kilo - (useBig * 5);  
    return remaining <= small;  
}  
}
```

OUTPUT:

```
D:\bmsce\2sem\Java programming\4week>java Ricebags  
Enter the number of small rice bags (1 kilo each): 3  
Enter the number of big rice bags (5 kilo each): 4  
Enter the kilos of rice: 22  
It is possible to pack the rice.  
  
D:\bmsce\2sem\Java programming\4week>java Ricebags  
Enter the number of small rice bags (1 kilo each): 2  
Enter the number of big rice bags (5 kilo each): 2  
Enter the kilos of rice: 15  
It is NOT possible to pack the rice.
```

2. Write a class called Employee, which models an employee with an ID, name and salary, is designed as shown in the following class diagram. The method `raiseSalary(percent)` increases the salary by the given percentage. Write the Employee class.



PROGRAM:

```

class Employee {
    private int id;

    private String firstName;

    private String lastName;

    private int salary;

    Employee(int id, String firstName, String lastName, int salary) {
        this.id = id;

        this.firstName = firstName;

        this.lastName = lastName;

        this.salary = salary;}

    public int getId() {
        return id;}

    public String getFirstName() {
        return firstName;
  
```

```

    }

    public String getLastName() {
        return lastName;
    }

    public String getName() {
        return (firstName + " " + lastName);
    }

    public int getSalary() {
        return salary;
    }

    public void setSalary(int salary) {
        this.salary = salary;
    }

    public int getAnnualSalary() {
        return salary * 12;
    }

    int raiseSalary(int percent) {
        return this.salary = this.salary + (this.salary * percent / 100);
    }

    public String toString() {
        return "Employee[id=" + id + ",name=" + getName() + ",salary=" + salary + "];"
    }
}

public class EmployeeMain {
    public static void main(String[] args) {
        Employee emp = new Employee(1, "puneeth", "kumar", 15000);
        System.out.println("Employee id:"+emp.getId());
    }
}

```

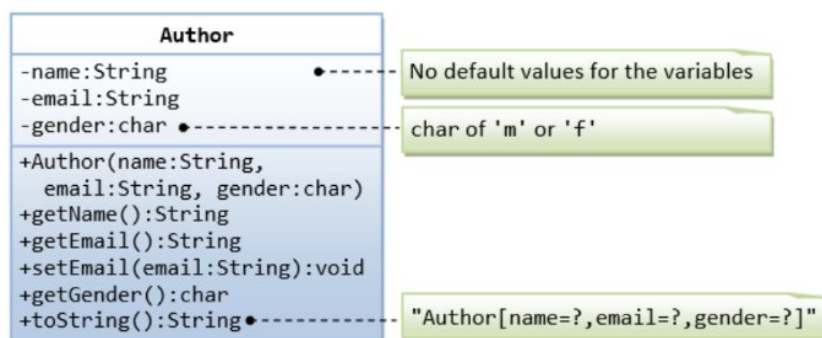
```
System.out.println("Employee firstname:"+emp.getFirstName());  
System.out.println("Employee lastname:"+emp.getLastName());  
System.out.println("Employee fullname:"+emp.getName());  
System.out.println("Employee salary:"+emp.getSalary());  
System.out.println("Annual Salary: " + emp.getAnnualSalary());  
emp.setSalary(20000);  
System.out.println("New salary of employee:"+emp.getSalary());  
emp.raiseSalary(30);  
System.out.println("New salary after 30% raise:"+emp.getSalary());  
System.out.println("Employee details: " + emp.toString());  
}  
}
```

OUTPUT:

```
D:\bmsce\2sem\Java programming\4week>java EmployeeMain  
Employee id:1  
Employee firstname:puneeth  
Employee lastname:kumar  
Employee fullname:puneeth kumar  
Employee salary:15000  
Annual Salary: 180000  
New salary of employee:20000  
New salary after 30% raise:26000  
Employee details: Employee[id=1,name=puneeth kumar,salary=26000]
```

3. Write a class called **Author** (as shown in the class diagram) is designed to model a book's author. It contains:

- Three private instance variables: name (String), email (String), and gender (char of either 'm' or 'f');
- One constructor to initialize the name, email and gender with the given values; **public Author (String name, String email, char gender) {.....}**
- public getters/setters: **getName()**, **getEmail()**, **setEmail()**, and **getGender()**;
- A **toString()** method that returns "Author[name=?,email=?,gender=?]", e.g., "Author[name=Tan Ah Teck,email=ahTeck@somewhere.com,gender=m]".



PROGRAM:

```
class Author {  
  
    private String name;  
  
    private String email;  
  
    private char gender;  
  
    Author(String name, String email, char gender) {  
  
        this.name = name;  
  
        this.email = email;  
  
        this.gender = gender;}  
  
    public String getName() {  
  
        return this.name;}  
  
    public String getEmail() {  
  
        return this.email;}  
  
    public void setEmail(String email) {
```

```

        this.email = email; }

    public char getGender() {

        return this.gender;}

    public String toString() {

        return ("Author[name=" + this.name + ", email=" + this.email + ", gender=" +
this.gender + "]");

    }}

class AuthorMain {

    public static void main(String args[]) {

        Author a1 = new Author("robert", "author@book.com", 'm');

        System.out.println("Author name:" + a1.getName());

        System.out.println("Author email:" + a1.getEmail());

        System.out.println("Author gender:" + a1.getGender());

        a1.SetEmail("author@gmail.com");

        System.out.println("Author new email:" + a1.getEmail());

        System.out.println("Author details:" + a1);

    }

}

```

OUTPUT:

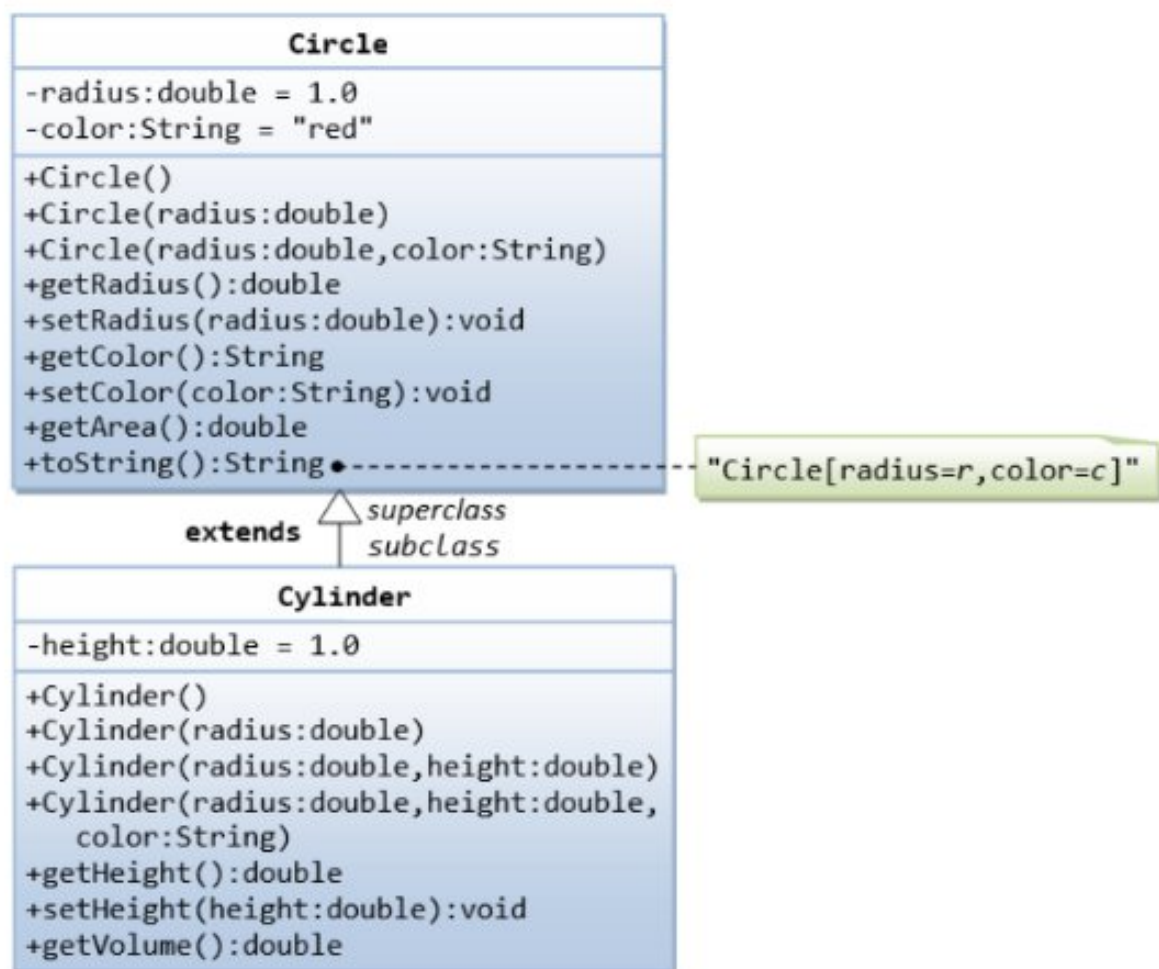
```

D:\bmsce\2sem\Java programming\4week>java AuthorMain
Author name:robert
Author email:author@book.com
Author gender:m
Author new email:author@gmail.com
Author details:Author[name=robert, email=author@gmail.com, gender=m]

```

4. In this exercise, a subclass called Cylinder is derived from the superclass Circle as shown in the class diagram. Study how the subclass Cylinder invokes the superclass' constructors (via `super()` and `super(radius)`) and inherits the variables and methods from the superclass Circle.

- The subclass Cylinder inherits `getArea()` method from its superclass Circle. Try overriding the `getArea()` method in the subclass Cylinder to compute the surface area ($=2\pi \times \text{radius} \times \text{height} + 2 \times \text{base-area}$) of the cylinder instead of base area.
- That is, if `getArea()` is called by a Circle instance, it returns the area.
- If `getArea()` is called by a Cylinder instance, it returns the surface area of the cylinder.
- If you override the `getArea()` in the subclass Cylinder, the `getVolume()` no longer works. This is because the `getVolume()` uses the overridden `getArea()` method found in the same class. Fix the `getVolume()`.



PROGRAM:


```
class Circle {  
    private double radius;  
    private String color = "red";  
    Circle() {  
        this.radius = 1.0;  
        this.color = "blue";  
    }  
    Circle(double radius) {  
        this.radius = radius;  
    }  
    Circle(double radius, String color) {  
        this.radius = radius;  
        this.color = color;  
    }  
    public double getRadius() {  
        return this.radius;  
    }  
    public void setRadius(double radius) {  
        this.radius = radius;  
    }  
    public String getColor() {  
        return this.color;  
    }  
    public void setColor(String color) {  
        this.color = color;  
    }  
    public double getArea() {
```

```

        return 3.14 * this.radius * this.radius;
    }
    public String toString() {
        return ("Circle [radius=" + this.radius + ", color=" + this.color
            + "]");
    }
}

class Cylinder extends Circle {
    double height;

    Cylinder() {
        super();
        this.height = 1.0;
    }

    Cylinder(double radius, double height) {
        super(radius);
        this.height = height;
    }

    Cylinder(double radius, double height, String color) {
        super(radius, color);
        this.height = height;
    }

    public double getHeight() {
        return this.height;
    }

    public void setHeight(double height) {
        this.height = height;
    }
}

```

@Override

```
public double getArea() {  
    double baseArea = super.getArea();  
    double sideArea = 2 * 3.14 * getRadius() * height;  
    return 2 * baseArea + sideArea;  
}  
  
public double getVolume() {  
    return super.getArea() * height;  
}  
}  
  
class CircleCylinder {  
    public static void main(String[] args) {  
        Cylinder c = new Cylinder(2.0, 5.0, "blue");  
        System.out.println(c);  
        System.out.println("Cylinder radius:" + c.getRadius());  
        c.setRadius(3.0);  
        System.out.println("new radius:" + c.getRadius());  
        System.out.println("Volume:" + c.getVolume());  
        System.out.println("Area:" + c.getArea());  
        System.out.println("Height:" + c.getHeight());  
        c.setHeight(4.0);  
        System.out.println("New Height:" + c.getHeight());  
        System.out.println("Color:" + c.getColor());  
        c.setColor("green");  
        System.out.println("new color:" + c.getColor());  
        System.out.println("Volume = " + c.getVolume());  
        System.out.println("Cylinder details:" + c);  
    }  
}
```

```
}  
}
```

OUTPUT:

```
D:\bmsce\2sem\Java programming\4week>java CircleCylinder  
Circle [radius=2.0, color=blue]  
Cylinder radius:2.0  
new radius:3.0  
Volume:141.29999999999998  
Area:150.72  
Height:5.0  
New Height:4.0  
Color:blue  
new color:green  
Volume = 113.03999999999999  
Cylinder details:Circle [radius=3.0, color=green]
```

5. Define a class CARRENTAL with the following details :

- **Class Members are:** CarId of int type, CarType of string type and Rent of float type.
- **Define GetCar()** method which accepts CarId and CarType.
- **GetRent()** method which return rent of the car on the basis of car type, i.e.
Small Car= 1000, Van = 800, SUV = 2500
- **ShowCar()** method which allow user to view the contents of cars i.e. id, type and rent.

PROGRAM:

```
import java.util.Scanner;

class CARRENTAL {

    int CarId;

    String CarType;

    float Rent;

    void GetCar(int id, String type) {

        CarId = id;

        CarType = type;

        Rent = GetRent(); }

    float GetRent() {

        if (CarType.equalsIgnoreCase("Small Car")) {return 1000; }

        else if (CarType.equalsIgnoreCase("Van")) {return 800; }

        else if (CarType.equalsIgnoreCase("SUV")) {return 2500; }

        else { return 0; } }

    void ShowCar() {

        System.out.println("Car ID: " + CarId);

        System.out.println("Car Type: " + CarType);

        System.out.println("Rent: " + Rent);

    } }
```

```

public class CarRentalDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        CARRENTAL car = new CARRENTAL();
        System.out.print("Enter Car ID: ");
        int id = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter Car Type (1:Small Car / 2:Van / 3:SUV)only number: ");
        String type = sc.nextLine();
        car.GetCar(id, type);
        System.out.println("\nCar Details:");
        car.ShowCar();
    }
}

```

OUTPUT:

```

D:\bmsce\2sem\Java programming\4week>java CarRentalDemo
Enter Car ID: 1
Enter Car Type (1:Small Car / 2:Van / 3:SUV)only number: small car

Car Details:
Car ID: 1
Car Type: small car
Rent: 1000.0

D:\bmsce\2sem\Java programming\4week>java CarRentalDemo
Enter Car ID: 2
Enter Car Type (1:Small Car / 2:Van / 3:SUV)only number: suv

Car Details:
Car ID: 2
Car Type: suv
Rent: 2500.0

```