

CSE1007-JAVA PROGRAMMING-LAB  
EXERCISE-02

Name: Kulvir Singh  
Reg. No.: 19BCE2074

**Question 1:**

The abstract Widget class has four concrete subclasses named WidgetA, WidgetB, WidgetC and WidgetD. Each of these four classes has a different mass in kilograms. The mass of any WidgetA object is 4 kilograms. The masses for WidgetB, WidgetC and WidgetD classes are 1, 5, and 17 kilograms, respectively. Each Widget object has a string that identifies its color. Create six different widgets, and store them in a one-dimension array. Display the entries in the array and their total mass.

**Aim:**

To create a Widget class that has four subclasses. The subclasses should have different mass values and a string to store the color. Enter the values of mass to the widget then access through objects which send the value of color. Display the sum of total mass stored in each Widget.

**Algorithm:**

1. Start
2. Create a Widget class that defines the structure of its subsidiary classes.
3. Create 2 data members of mass(integer) and color(characters)
4. Create 4 subclasses which extend the class Widget.
5. Each of the subclasses also have a Constructor with parameter to initialize the value of color.
6. Use the super class data member definition to initialize the data members.
7. Create the main method to run the code.
8. Take 4 user inputs for colors.
9. Create 4 objects of the subclasses(1 each).
10. Each object sends a single variable which contains the color input by the user.
11. Initialize an array of size 4
12. Store the value of mass by from each subclass by using object to call the data member of the class
13. Calculate the sum of the array elements using loops
14. Display each of the inputs taken in the main method for the color.
15. Display each array element
16. Display the sum of the array elements.
17. Stop

**Code:**

```
import java.util.*;
abstract class Widget{
    int mass;
    String color;
}
```

```

class WidgetA extends Widget{
    WidgetA(String c){
        super.mass=4;
        super.color=c;
    }
}
class WidgetB extends Widget{
    WidgetB(String c){
        super.mass=1;
        super.color=c;
    }
}
class WidgetC extends Widget{
    WidgetC(String c){
        super.mass=5;
        super.color=c;
    }
}
class WidgetD extends Widget{
    WidgetD(String c){
        super.mass=17;
        super.color=c;
    }
}
class AbstractWidgetMass{
    public static void main(String[] args) {
        Scanner x = new Scanner(System.in);
        String a,b,c,d;
        System.out.println("Enter 4 colors :");
        a=x.next();
        b=x.next();
        c=x.next();
        d=x.next();
        WidgetA o1 = new WidgetA(a);
        WidgetC o3 = new WidgetC(c);
        WidgetB o2 = new WidgetB(b);
        WidgetD o4 = new WidgetD(d);
        int arr[] = new int[4];
        arr[0]=o1.mass;
        arr[1]=o2.mass;
        arr[2]=o3.mass;
        arr[3]=o4.mass;
        int s=0;
        for(int i=0;i<4;i++){
            System.out.print(arr[i]);
            s=s+arr[i];
            System.out.println();
        }
    }
}

```

```
        System.out.println("Sum of masses = "+s);  
    }  
}
```

### Output:

```
C:\Users\kulvir\Desktop\JAVA CODEs>javac AbstractWidgetMass.java  
C:\Users\kulvir\Desktop\JAVA CODEs>java AbstractWidgetMass  
Enter 4 colors :  
Red  
Blue  
Green  
Black  
4  
1  
5  
17  
Sum of masses = 27  
C:\Users\kulvir\Desktop\JAVA CODEs>
```

## Question 2

Write a java program to implement the concept of Multilevel Inheritance to calculate the average marks of the student. Create the Parent Class named student, extend the child class Marks from student class to instantiate the data members namely, StuID, name. Create user defined display() method to display the results. Also create the third class average which extends student and Marks class to find the average for 5 subjects of decimal type.

### Aim:

To calculate the average marks of a student and display the student details by implementing the concept of Multilevel Inheritance in java.

### Algorithm:

Student class

1. Start
2. Declare global variables StudID and name of String type
3. Create a parameterized constructor that initializes StuID and name
4. Create method display
5. Display StuID and name
6. Stop

Marks class inherits Student class

1. Start
2. Declare global variable sum which is double type
3. Create a parameterized constructor
4. Constructor initiates data members of inherited class Student
5. Constructor initiates data member sum
6. Create method display and print the sum value
7. Stop

Average class inherits Marks class

1. Start
2. Declare global variable average which is double type
3. Create a parameterized constructor
4. Constructor initiates data members of inherited class Marks
5. Calculate the average using formula  $\text{average} = \text{sum} / \text{total number of subjects}$
6. Create method display and print the average value
7. Stop

Driver class

1. Start

2. Take user input of name and student id
3. Take user input of total
4. Create object of the Average class and use it to print the details and average of student
5. Stop

**Code:**

```
import java.util.*;

class Student{

    String StuID,name;

    Student(String id,String n){

        StuID = id;

        name =n;

    }

    void display(){

        System.out.println("Student ID : "+StuID);

        System.out.println("Student Name : "+name);

    }

}

class Marks extends Student{

    float sum;

    Marks(String s, String n,float total){

        super(s,n);

        sum=total;

    }

    void display1(){

        display();

        System.out.println("Total Marks = "+sum);

    }

}

class Average extends Marks{
```

```

float avg;
Average(String s,String n,float total)
{
    super(s,n,total);
    avg = (float)(total/5.0);
}
void display2(){
    display1();
    System.out.println("Average = "+avg);
}
}

class DriverInheritStudent{
    public static void main(String[] args) {
        Scanner x = new Scanner (System.in);
        System.out.println("Enter the Student ID and Name:");
        String a,b;
        a=x.next();
        b=x.next();
        System.out.println("Enter the total marks of the student:");
        float t = x.nextFloat();
        Average obj = new Average(a,b,t);
        obj.display2();
    }
}

```

**Output:**

```
C:\Users\kulvir\Desktop\JAVA CODEs>java DriverInheritStudent
Enter the Student ID and Name:
19BCE2074
Kulvir
Enter the total marks of the student:
498.234
Student ID : 19BCE2074
Student Name : Kulvir
Total Marks = 498.234
Average = 99.646805
```

### Question 3:

Create a package *shape* which can have classes *circle* and *square*. These classes can have required data members and methods to calculate the areas of circle and square. Create another class *calculate* which will use the above package and to create objects. Finally print the calculated areas.

### Aim:

To create a package *shape* with classes *circle* and *square* that calculates the areas of circles and squares respectively. Use the package to find the area of a square and circle.

### Algorithm:

Creation of package:

Square class (scope = public)

1. Start
2. Declare *s* and *a* of type double and scope of private
3. Create a public constructor to initialize *s*
4. Create a function *area* that return double value
5. Calculate area of square and store in *a* using formula  $a = s*s$
6. Return *a*
7. Stop

Circle class (scope = public)

1. Start
2. Declare *r* and *a* of type double and scope of private
3. Create a public constructor to initialize *r*
4. Create a function *area* that return double value
5. Calculate area of square and store in *a* using formula  $a = \pi*r*r$
6. Return *a*
7. Stop

Usage of Package:

1. Start
2. Import the 2 classes of package *shape* into the program
3. Create a class *calculate*
4. Create an object of *Square* class and send the value 77.23
5. Create an object of *Circle* class and send the value 23.6
6. Use the package function and display the areas
7. Stop

### Code:



### Creating package shape

```
package shape;
public class square{
    private double s;
    private double a;
    public square(double x){
        s=x;
    }
    public double area(){
        double a;
        a=s*s;
        return a;
    }
}

package shape;
public class circle{
    private double r;
    private double a;
    public circle(double x){
        r=x;
    }
    public double area(){
        double a;
        a=Math.PI*r*r;
        return a;
    }
}
```

### Using package shape

```
import shape.circle;
import shape.square;
class calculate{
    public static void main(String[] args) {
        circle c = new circle(23.6);
        System.out.println("Area of circle of radius 23.6 is "+c.area());
        square s = new square(77.23);
        System.out.println("Area of square of side 77.23 is "+s.area());
    }
}
```

**Output:**

```
C:\Users\kulvir\Desktop\JAVA CODES\my-packages>java calculate
Area of circle of radius 23.6 is = 1749.7414443433715
Area of square of side 77.23 is = 5964.472900000001
```

#### Question 4:

Create class Car with below attributes:

- make
- model
- passengerCapacity
- onRoadPrice

Create class CarDemo which has main method. Create another method - getBestCar in this class which will take four car objects and additional parameter as compareType. Value of this parameter can be 'c' or 'p', where 'c' stands for capacity of passengers and 'p' stands for on road price. This method will return make and model name for highest value of 'c' or 'p' (with – character in between. E.g for make – Hyundai and model – santro, method should return “Hyundai-santro”). Create four objects of Car class in main method and drive it through getBestCar method.

#### Aim:

To get the best car out of given set of cars depending on the on road price and car capacity with the use of inheritance in Java

#### Algorithm:

1. Start
2. Create a class named Car
3. Declare data members make, model as String
4. Declare data members passengerCapacity and onRoadPrice as int and long.
5. Create a parameterized constructor to initialize the data members
6. Create a method named get best car with parameters as 4 objects and one char
7. Create an object array and store the 4 object parameters in the array
8. Check if the value of char parameter is 'c' then follow the step 10 to 12
9. Else if the value of char parameter is 'p' then follow step 13 to
10. Set the value of max to =0
11. Compare the value of passengerCapacity of each object and store the highest value into max
12. Display the Car name – Car model of the object that contains the max
13. Set the value of max to =0
14. Compare the value of onRoadPrice of each object and store the highest value into max
15. Display the Car name – Car model of the object that contains the max
16. Stop

Driver class algorithm:

1. Start
2. Create a class CarDemo
3. Create main method

4. Create 4 objects of Car class with proper parameters
5. Create an object o of the Car class
6. Use o to get the best car
7. Stop

**Code:**

```
class Car{
    String make;
    String model;
    int passengerCapacity;
    long onRoadPrice;
    Car(String a,String b,int c,long d){
        make = a;
        model = b;
        passengerCapacity = c;
        onRoadPrice = d;
    }
    Car(){ }
    String getBestCar(Car w,Car x,Car y,Car z,char compareType){
        Car o[] = new Car[4];
        o[0]=w;o[1]=x;o[2]=y;o[3]=z;
        if(compareType=='c'){
            int max=0;int flag=0;
            for(int i=0;i<4;i++){
                if(o[i].passengerCapacity>max){
                    max=o[i].passengerCapacity;
                    flag=i;
                }
            }
            String s = o[flag].make+"-"+o[flag].model;
```

```

        return s;
    }
    else{
        long max1=0;int flag1=0;
        for(int i=0;i<4;i++){
            if(o[i].onRoadPrice>max1){
                max1=o[i].onRoadPrice;
                flag1=i;
            }
        }
        String s1 = o[flag1].make+"-"+o[flag1].model;
        return s1;
    }

}

}

}

class CarDemo{
    public static void main(String[] args) {
        Car o1 = new Car("Hyundai","Creta",5,1600000);
        Car o2 = new Car("Honda","City",4,2300000);
        Car o3 = new Car("Ford","EcoSport",6,1300000);
        Car o4 = new Car("Nissan","Sunny",4,1600000);
        Car o = new Car();
        System.out.println("Best Car according to Person Capacity \n"+o.getBestCar(o1, o2, o3, o4, 'c'));
        System.out.println("Best Car according to On Road Price \n"+o.getBestCar(o1, o2, o3, o4, 'p'));
    }
}

```

**Output:**

```
C:\Users\kulvir\Desktop\JAVA CODEs>java CarDemo
Best Car according to Person Capacity
Ford-EcoSport
Best Car according to On Road Price
Honda-City
C:\Users\kulvir\Desktop\JAVA CODEs>
```