

## Week4 – Batch 3

---

Monday, December 27, 2021 2:19 AM

**1.Create Area class with variable height. Create a triangle class that extends area class with variables base and method to calculate the area. Create a rectangle class that extends area class with variable width and method to calculate area. Now create a triangle and rectangle objects and print their areas.**

```
import java.util.Scanner;
import java.util.InputMismatchException;

class Area{
    Scanner sc = new Scanner(System.in);
    double height;
    void Height() {
        System.out.println("Enter Height: ");
        height = sc.nextDouble();
    }
}

class Triangle extends Area{
    Scanner sc = new Scanner(System.in);
    double base, Tarea;
    void calArea() {
        System.out.println("Inside Triangle Class, get calArea");
        System.out.println("Enter Base: ");
        base = sc.nextDouble();
        Tarea = (double)0.5*base*height;
        System.out.println("Area of Triangle: "+ Tarea);
        if(base<= 0 || height<=0) {
            throw new InputMismatchException("Enter Value more than 0");
        }
    }
}

class Rectangle extends Area{
    Scanner sc = new Scanner(System.in);
    double length,width, Rarea;
    void calArea() {
        System.out.println("Inside Rectangle Class, get calArea");
        System.out.println("Enter Length: ");
        length = sc.nextDouble();
        System.out.println("Enter Width: ");
        width = sc.nextDouble();
        Rarea = (double)length*width;
        System.out.println("Area of Triangle: "+ Rarea);
        if(width<= 0 || length<=0) {
            throw new InputMismatchException("Enter Value more than 0");
        }
    }
}

import java.util.Scanner;

public class Run {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Triangle T = new Triangle();
        Rectangle R = new Rectangle();
    }
}
```

```

T.Height();
T.calArea();
R.Height();
R.calArea();

}
}

```

### Output

Tasks:

- 1.Raise exception for negative input value.
- 2.Display the concept of Inheritance.
- 3.Use keywords this, final and static for variables
- 4.Show minimum 2 testcases of your own:
  - a)Correct b)Wrong

#### 4 a)

```

Enter Height:
5
Inside Triangle Class, get calArea
Enter Base:
5
Area of Triangle: 12.5
Enter Height:
5
Inside Rectangle Class, get calArea
Enter Length:
-5
Enter Width:
5
Area of Triangle: -25.0
Exception in thread "main"
java.util.InputMismatchException: Enter
Value more than 0

```

#### 4 b)

```

Enter Height:
5
Inside Triangle Class, get calArea
Enter Base:
5
Area of Triangle: 12.5
Enter Height:
5
Inside Rectangle Class, get calArea
Enter Length:
5
Enter Width:
5
Area of Triangle: 25.0

```

2. Create two classes as Vehicle and Car where car is inheriting vehicle. Now create an object for the car and print the details. Note that data member and methods can be defined in both classes.

### Output

```

import java.util.Scanner;

class Vehicle{
Scanner sc = new Scanner(System.in);
String Name;
String Model;
void getInfo1() {
System.out.println("Inside Vehicle, getInfo1");
System.out.println("Enter Name: ");
Name = sc.nextLine();
System.out.println("Enter Model: ");
Model= sc.nextLine();
}
}

class Car extends Vehicle {
Scanner sc= new Scanner(System.in);
String Color;
double Price;
void getInfo2() {
System.out.println("Inside Car, getInfo2");

```

Tasks:

- 1.Display the concept of Inheritance.
- 2.Provide at least four variables for each car and print them.
- 3.Show If user enters two car with the same name, raise an exception.
- 4.Show at least 4 different car data.

```

Enter No. of Cars:
4
Inside Vehicle, getInfo1
Enter Name:
Mercedes
Enter Model:
AMG
Inside Car, getInfo2
Enter Color:
Black
Enter Price:
10000

Inside Vehicle, getInfo1
Enter Name:

```

```

Inside Car, Display
Details:
Name: Mercedes
Model: AMG
Color: Black
Price: 10000.0

```

```

Inside Car, Display

```

```

System.out.println("Enter Color: ");
Color = sc.nextLine();
System.out.println("Enter Price: ");
Price = sc.nextDouble();
}

void display() {
System.out.println("Inside Car, Display");
System.out.println("Details: ");
System.out.println("Name: " + Name);
System.out.println("Model: " + Model);
System.out.println("Color: " + Color);
System.out.println("Price: " + Price);
}
}

import java.util.Scanner;

public class Run2 {

public static void main(String[] args) {
// TODO Auto-generated method stub
Car[] c = new Car[10];
Scanner sc= new Scanner(System.in);
System.out.println("Enter No. of Cars: ");
int n = sc.nextInt();
for(int i = 0; i<n; i++) {
c[i] = new Car();
c[i].getInfo1();
c[i].getInfo2();
c[i].display();}

try {
if(n>=2) {
for(int j = 1; j<n; j++) {
if((c[j].make).equals(c[j-1].make)){
throw new IllegalArgumentException();
}
}
}
}
catch (Exception e) {
System.out.println(e + " Two Cars are Same");
}

}

}
}

```

```
BMW
Enter Model:
Coupe
Inside Car, getInfo2
Enter Color:
Orange
Enter Price:
12000
Inside Vehicle, getInfo1
Enter Name:
AUDI
Enter Model:
R8
Inside Car, getInfo2
Enter Color:
White
Enter Price:
15000
Inside Vehicle, getInfo1
Enter Name:
Mercedes
Enter Model:
AMG
Inside Car, getInfo2
Enter Color:
Silver
Enter Price:
10000
```

```

Details:
Name: BMW
Model: Coupe
Color: Orange
Price: 12000.0

Inside Car, Display
Details:
Name: AUDI
Model: R8
Color: White
Price: 15000.0

Inside Car, Display
Details:
Name: Mercedes
Model: AMG
Color: Silver
Price: 10000.0

IllegalArgumentOutOfRangeException
Two Cars are Same

```

**3. In a single package demonstrate default, public, protected, and private access modifiers using Inheritance concepts.**

```
import java.util.Scanner;
class A {
    private int i = 10;
    protected String s = "This is Protected";
}

private class P {
    private double d = 28.09;
    public String s1 = "Public string in private class"; //error
}

protected class B {
```

```

public int i1 = 20;    //error
}
class X extends A {
}
public class Q3 {
    protected double p1 = 2.5;
    public static void main(String[] args) {
        Scanner scan = new Scanner (System.in);
        X x = new X();
        System.out.println(x.s);
        System.out.println(s);    //gives an error
        Q3 am = new Q3();
        System.out.println(am.p1);
    }
}

```

```

public class Q3pt2 {

    public static void main(String[] args) {
        A a = new A();
        //System.out.println(i); //gives error
        System.out.println(a.s);
        X x1 = new X();
        Q3 q = new Q3();
        System.out.println(q.p1);
    }
}

```

### Output

- Tasks:1.Create objects for each below class and document the execution.  
2.Create four classes:

a)Public class with protected variables					2.5
---	--	--	--	--	-----

b)Default class with private and protected variables				This is Protected
--	--	--	--	-------------------

c)Private class private and public variables (error)				Error
--	--	--	--	-------

d)Protected class (public variables)					Error
--------------------------------------	--	--	--	--	-------

