NAME	**:**	**NITHISH G**
REG NO.	**:**	**19BCS0012**
COURSE	**:**	**JAVA PROGRAMMING**
DATE	**:**	**01.05.2021**

1. a) Multilevel inheritance calculator

Source code :

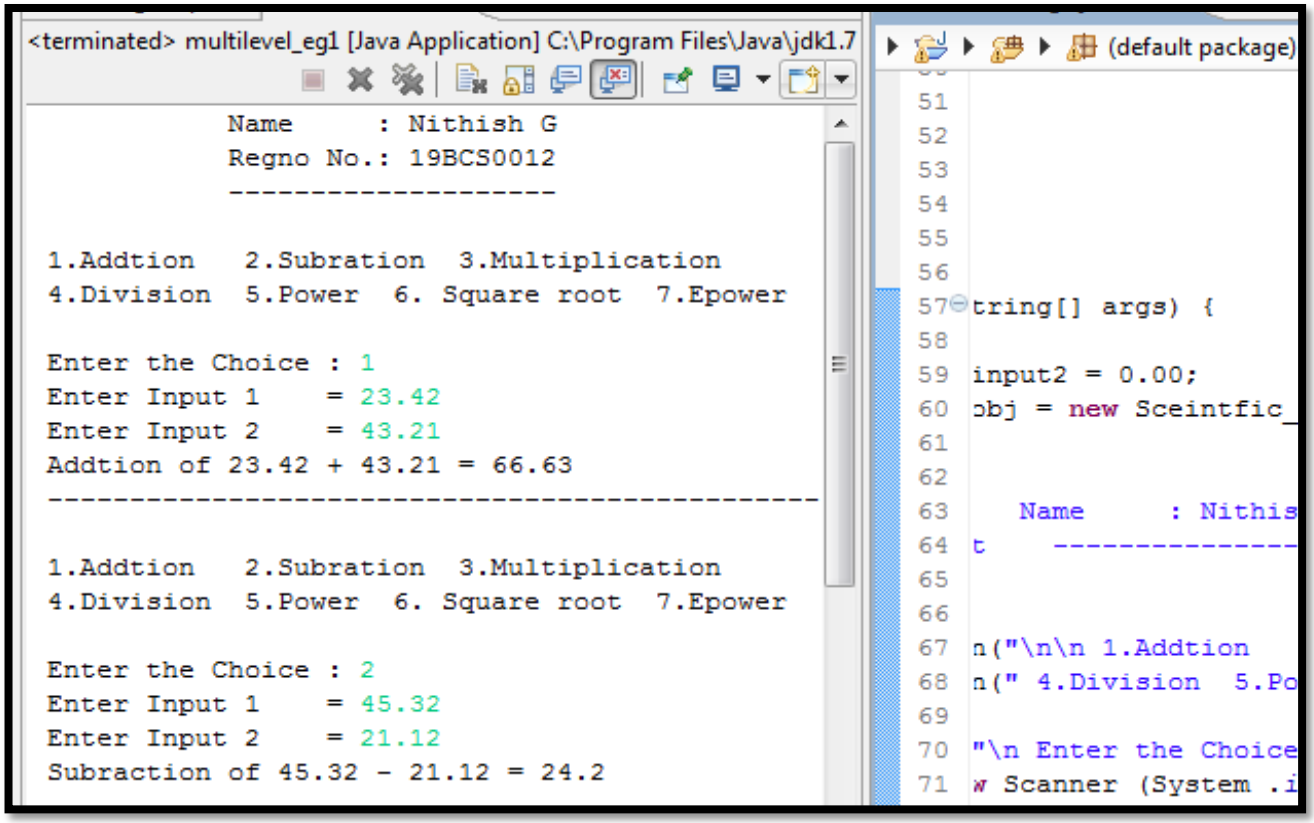
```
import java.lang.Math;
import java.util.Scanner;

class calculator
{
        double x  = 0.0,y =0.0, result =0.0;
        public void set_value(double a , double b)
        {
                x = a ;
                y = b ;
        }

        public void show_result()
        {
                System.out.println(result);
        }
}
class simple_calculator extends calculator
{
        public void add()
        {
         result = x + y ;
        }
        public void sub()
        {
         result = x - y ;
        }
        public void mul()
        {
         result = x * y ;
        }
```

```
        public void div()
        {
         result = x / y ;
        }
}
class Sceintfic_calculator extends simple_calculator{

        public void pow()
        {
                result = Math.pow(x, y);
        }
        public void sqrt()
        {
                result = Math.sqrt(x);
        }
        public void epow()
        {
                result = Math.exp(x);
        }


}

public class multilevel_eg1 {

        public static void main(String[] args) {

                double input1=0.00 , input2 = 0.00;
                Sceintfic_calculator obj = new Sceintfic_calculator();

                int choice = 0 ;
                System.out.print("\t    Name     : Nithish G \n\t    Regno No.: 19BCS0012\n");
                  System.out.print("\t    --------------------");
                while (choice!= 999)
                {
                        System.out.println("\n\n 1.Addtion   2.Subration  3.Multiplication");
                        System.out.println(" 4.Division  5.Power      6. Square  7.Epower");

                        System.out.print("\n Enter the Choice : ");
                        Scanner read = new Scanner (System .in);
                        choice = read .nextInt();
                        if(choice <= 5)
                        {
                                System.out.print(" Enter Input 1 \t  = ");
                                input1 = read.nextDouble();
```

```
                System.out.print(" Enter Input 2 \t  = ");
                input2 = read.nextDouble();
        }
        else
        {
                System.out.print(" Enter Input      : ");
                input1 = read.nextDouble();
        }
switch(choice)
{
case 1:
        obj.set_value(input1,input2);
        obj.add();
        System.out.print(" Addtion of "+ input1+" + " + input2 + " = ");
        obj.show_result();
        System.out.print(" ---------------------------------------------");
        break;
case 2:
        obj.set_value(input1,input2);
        obj.sub();
        System.out.print(" Subraction of "+ input1+" - " + input2 + " = ");
        obj.show_result();
        System.out.print(" ---------------------------------------------");
        break;
case 3:
        obj.set_value(input1,input2);
        obj.mul();
        System.out.print(" Multiplication of "+ input1+" x " + input2 + " = ");
        obj.show_result();
        System.out.print(" ---------------------------------------------");
        break;
case 4:
        obj.set_value(input1,input2);
        obj.div();
        System.out.print(" Division of "+ input1+" / " + input2 + " = ");
        obj.show_result();
        System.out.print(" ---------------------------------------------");
        break;

case 5:
        obj.set_value(input1,input2);
        obj.pow();
        System.out.print(" Power of "+ input1+" ^ " + input2 + " = ");
        obj.show_result();
        System.out.print(" ---------------------------------------------");
```

```
                break;
            case 6:
                obj.set_value(input1,0);
                obj.sqrt();
                System.out.print(" Squart root of "+ +input1+" = ");
                obj.show_result();
                System.out.print(" ----------------------------------------------");
                break;
            case 7:
                obj.set_value(input1,0);
                obj.epow();
                System.out.print(" EPower root of "+ input1 + " = ");
                obj.show_result();
                System.out.print(" ----------------------------------------------");
                break;

            default :
                System.out.println(" invaild Input .....");
                System.out.print(" ----------------------------------------------");

            }
        }

}
}
```

Output



```
Subraction of 45.32 - 21.12 = 24.2
-----------------------------------------------

1.Addtion   2.Subration  3.Multiplication
4.Division  5.Power  6. Square root  7.Epower

Enter the Choice : 3
Enter Input 1     = 4.12
Enter Input 2     = 5.2
Multiplication of 4.12 x 5.2 = 21.4240000000000
-----------------------------------------------
```

```
71 w Scanner (System .i
72 xtInt();
73
74
75 int(" Enter Input 1
76 .nextDouble();
77 int(" Enter Input 2
78 .nextDouble();
79
```

```
1.Addtion   2.Subration  3.Multiplication
4.Division  5.Power  6. Square root  7.Epower

Enter the Choice : 4
Enter Input 1     = 91.9
Enter Input 2     = 8
Division of 91.9 / 8.0 = 11.4875
-----------------------------------------------

1.Addtion   2.Subration  3.Multiplication
4.Division  5.Power  6. Square root  7.Epower

Enter the Choice : 5
Enter Input 1     = 2
Enter Input 2     = 6
Power of 2.0 ^ 6.0 = 64.0
-----------------------------------------------

1.Addtion   2.Subration  3.Multiplication
4.Division  5.Power  6. Square root  7.Epower

Enter the Choice : 6
Enter Input       : 2704
Squart root of 2704.0 = 52.0
-----------------------------------------------
```

```
57 tring[] args) {
58
59 input2 = 0.00;
60 obj = new Sceintfic_
61
62
63    Name      : Nithis
64 t     ---------------
65
66
67 n("\n\n 1.Addtion
68 n(" 4.Division  5.Po
69
70 "\n Enter the Choice
71 w Scanner (System .i
72 xtInt();
73
74
75 int(" Enter Input 1
76 .nextDouble();
77 int(" Enter Input 2
78 .nextDouble();
79
```

Problems | @ Javadoc | Declara

```
1.Addtion   2.Subration  3.Multiplication
4.Division  5.Power  6. Square root  7.Epower

Enter the Choice : 7
Enter Input       : 3
EPower root of 3.0 = 20.085536923187668
-----------------------------------------------

1.Addtion   2.Subration  3.Multiplication
4.Division  5.Power  6. Square root  7.Epower

Enter the Choice : 999
System Exit....
```

```
68 n(" 4.Division  5.Po
69
70 "\n Enter the Choice
71 w Scanner (System .i
72 xtInt();
73
74
75 int(" Enter Input 1
76 .nextDouble();
77 int(" Enter Input 2
78 .nextDouble();
79
```

Problems | @ Javadoc | Declar

1 b). **Hierarchical inheritance (Calculate Areas)**

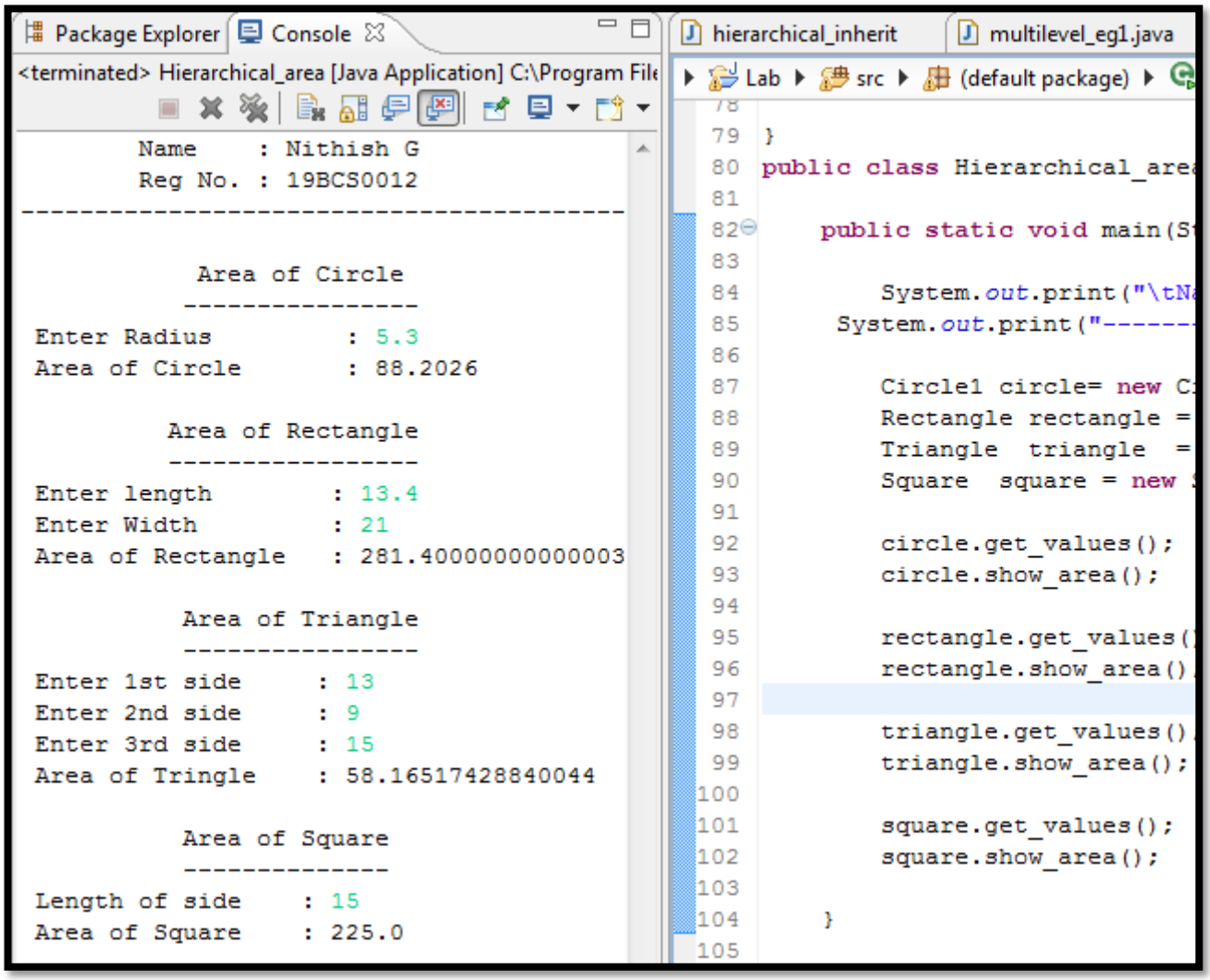
Source Code:

```
import java.util.Scanner;

abstract class graphic
{
        double input1 ,input2, input3;
        abstract void get_values();
        abstract void show_area();
        static Scanner read = new Scanner(System.in);
}
class Circle1 extends graphic
{

        void get_values()
        {
                System.out.println("            Area of Circle ");
                System.out.println("           ---------------- ");
                System.out.print(" Enter Radius        : ");
                this.input1 = read.nextDouble();
        }
         void show_area()
        {
                System.out.println(" Area of Circle       : " + ((3.14) * (input1 * input1))+"\n");

        }

}
class Rectangle extends graphic
{
        void get_values()
        {
                System.out.println("          Area of Rectangle ");
                System.out.println("          ----------------- ");
                System.out.print(" Enter length        : ");
                input1 = read.nextDouble();
                System.out.print(" Enter Width         : ");
                input2 = read.nextDouble();
        }
        void show_area()
        {
                System.out.print(" Area of Rectangle   : " + (input1*input2)+"\n\n");
        }
```

```
}
class Triangle extends graphic
{
        void get_values()
        {
                System.out.println("          Area of Triangle ");
                System.out.println("          ---------------- ");
                System.out.print(" Enter 1st side     : ");
                input1 = read.nextDouble();
                System.out.print(" Enter 2nd side     : ");
                input2 = read.nextDouble();
                System.out.print(" Enter 3rd side     : ");
                input3 = read.nextDouble();
        }
        void show_area()
        {
                double s = (input1+input2+input3)/2;

                System.out.print(" Area of Tringle    : " + Math.sqrt(s*((s-input1)*(s-input2)*(s-
input3)))+"\n\n");
        }

}
class Square extends graphic
{
        void get_values()
        {
                System.out.println("          Area of Square ");
                System.out.println("          -------------- ");
                System.out.print(" Length of side    : ");
                input1 = read.nextDouble();
        }
        void show_area()
        {
                System.out.print(" Area of Square    : " + (input1*input1)+"\n");
        }

}
public class Hierarchical_area {

        public static void main(String[] args) {

                System.out.print("\tName   : Nithish G \n\tReg No. : 19BCS0012\n");
         System.out.print("---------------------------------------------\n\n");
```

```
        Circle1 circle= new Circle1();
        Rectangle rectangle = new Rectangle();
        Triangle  triangle  = new Triangle();
        Square  square = new Square();

        circle.get_values();
        circle.show_area();

        rectangle.get_values();
        rectangle.show_area();

        triangle.get_values();
        triangle.show_area();

        square.get_values();
        square.show_area();

    }
}
```

Output



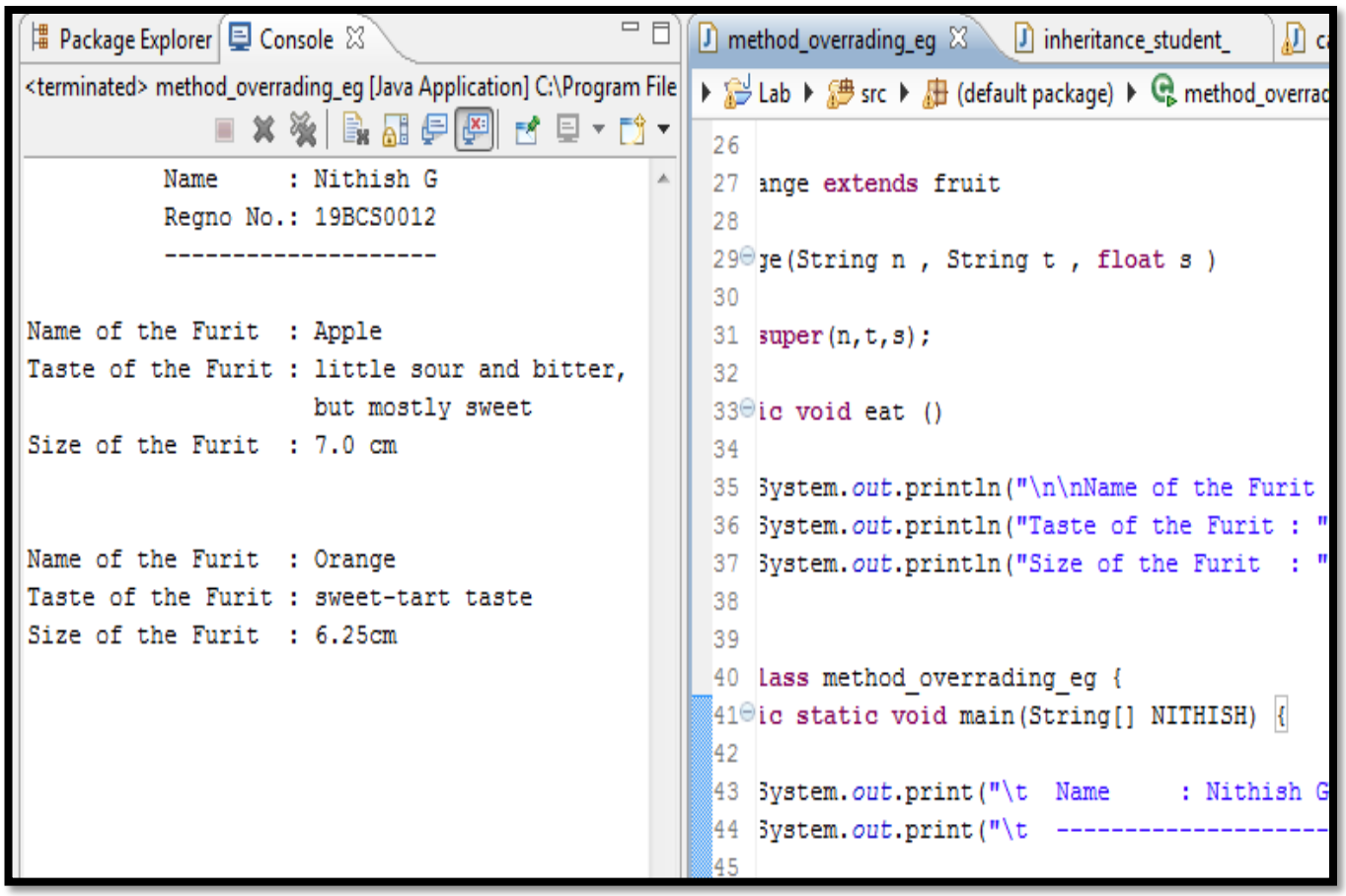
2. Create a base class Fruit with name, taste and size as its attributes. Create a method called eat() which describes the name of the fruit and its taste. Inherit the same in 2 other classes Apple and Orange and override the eat() method to represent each fruit taste.

Source Code

```
class fruit
{
        String fruit_name;
        String fruit_taste;
        float fruit_size;
        fruit(String name,String taste, float size)
        {
                fruit_name= name;
                fruit_taste= taste;
                fruit_size= size;
        }

}
class apple extends fruit
{
        apple(String n , String t , float s )
        {
                super(n,t,s);
        }
        public void eat ()
        {
                System.out.println("\n\nName of the Furit  : "+ fruit_name);
                System.out.println("Taste of the Furit : "+ fruit_taste);
                System.out.println("Size of the Furit  : "+ fruit_size+" cm");
        }
}
class Orange extends fruit
{
        Orange(String n , String t , float s )
        {
                super(n,t,s);
        }
        public void eat ()
        {
                System.out.println("\n\nName of the Furit  : "+ fruit_name);
                System.out.println("Taste of the Furit : "+ fruit_taste);
                System.out.println("Size of the Furit  : "+ fruit_size+"cm");
        }
}
```

```
public class method_overrading_eg {
        public static void main(String[] NITHISH) {

                System.out.print("\t  Name          : Nithish G \n\t  Regno No.: 19BCS0012\n");
                System.out.print("\t  --------------------");

        apple obj1 = new apple("Apple","little sour and bitter,\n\t\t     but mostly sweet" , 7);

                Orange obj2 = new Orange("Orange","sweet-tart taste" , 6.25f);
                obj1.eat();
                obj2.eat();
        }
}
```

Output

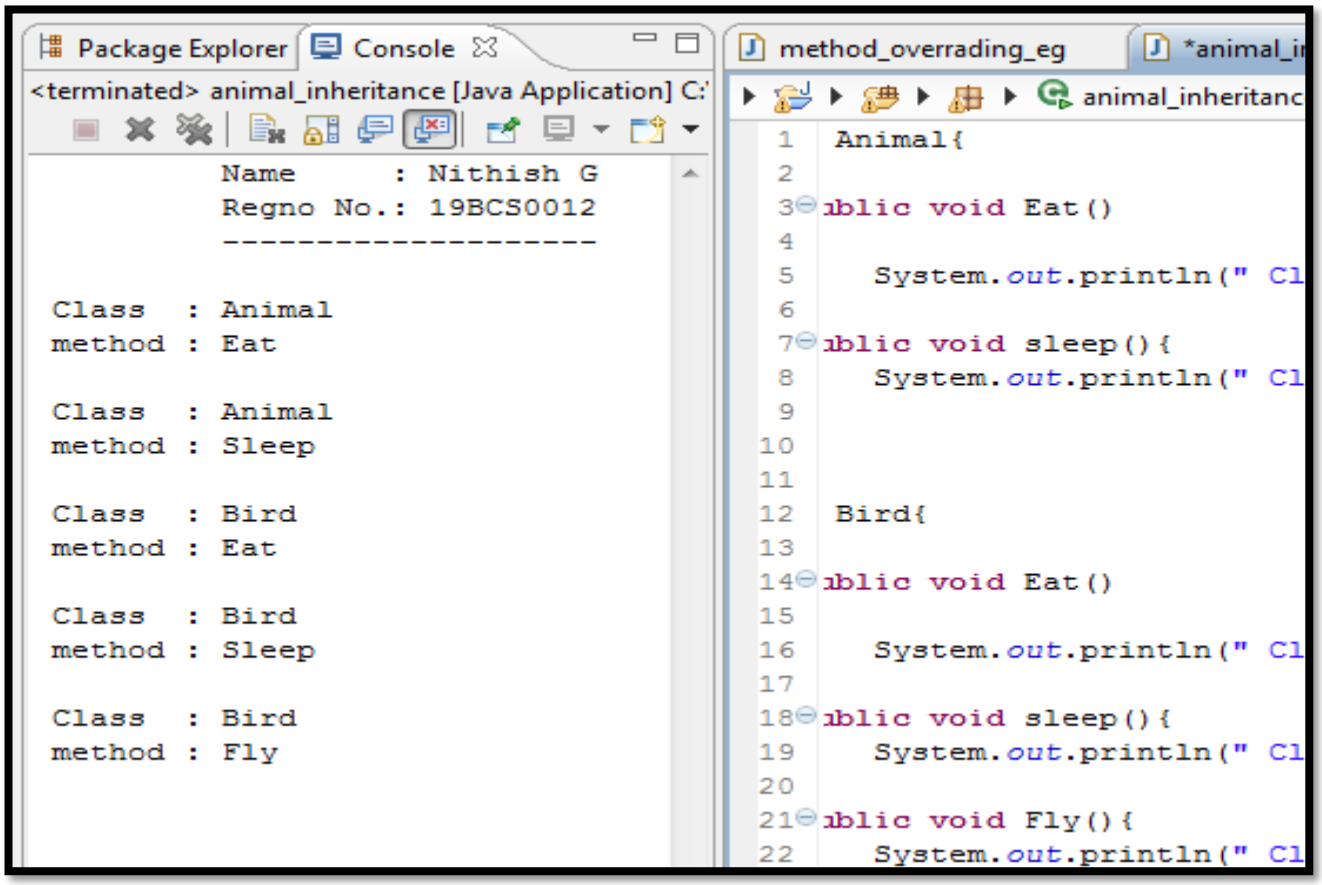


3. Create a class named 'Animal' which includes methods like eat() and sleep(). Create a child class of Animal named 'Bird' and override the parent class methods. Add a new method named fly(). Create an instance of Animal class and invoke the eat and sleep methods using this object. Create an instance of Bird class and invoke the eat, sleep and fly methods using this object.

Source Code:

```java
class Animal{

        public void Eat()
        {
                System.out.println(" Class  : Animal \n method : Eat \n");
        }
        public void sleep(){
                System.out.println(" Class  : Animal \n method : Sleep \n");
        }

}
class Bird{

        public void Eat()
        {
                System.out.println(" Class  : Bird \n method : Eat \n");
        }
        public void sleep(){
                System.out.println(" Class  : Bird \n method : Sleep \n");
        }
        public void Fly(){
                System.out.println(" Class  : Bird \n method : Fly \n");
        }

}
 public class animal_inheritance
{

        public static void main(String[] args) {
                System.out.print("\t  Name          : Nithish G \n\t  Regno No.: 19BCS0012\n");
                System.out.print("\t  --------------------\n\n");
                Animal ob1 = new Animal();
                Bird   ob2 = new Bird();

                ob1.Eat();
                ob1.sleep();
                ob2.Eat();
                ob2.sleep();
                ob2.Fly();
        }
}
```

Output



4. Create a superclass, Student, and two subclasses, Undergrad and Grad. The superclass Student should have the following data members: name, ID, grade, age, and address.The superclass, Student should have one method: boolean isPassed (double grade) The purpose of the isPassed method is to take one parameter, grade (value between 0 and 100) and check whether the grade has passed the requirement for passing a course.In the Student class this method should be empty as an abstract method. The two subclasses: Grad and Undergrad, wil inherit all data members of the Student class and override the method isPassed. For the UnderGrad class, if the grade is above 70.0, then isPassed returns true, otherwise it returns false. For the Grad class, if the grade is above 80.0, then isPassed returns true, otherwise returns false. Create a test class for your three classes. In the test class, create one Grad object and one Undergrad object. For each object, provide a grade and display the results of the isPassed method.

Source Code

```
abstract class Student {

        String student_name = "",student_address = "";
        int student_ID ,student_age ;
        double student_grade;

        abstract boolean isPassed (double grade);
```

```
}
class Undergrad extends Student  {

        Undergrad(String name ,int id, int age,  String addr)
        {

                student_name = name ;
                student_ID = id;
                student_age = age;
                student_address= addr;

                System.out.println("Student_Name      :  " +student_name   );
                System.out.println("Student_ID        :  " +student_ID   );
                System.out.println("Student_Age       :  " +student_age   );
                System.out.println("Student_Address   :  " +student_address   );

        }

String result = "";

        public boolean isPassed ( double grade )
        {
                student_grade = grade;
                System.out.println("Student_grad      :  " +student_grade   );

        if (student_grade>70)
        {
                return true;
        }
        else

                        return false;
        }

}
class Grad extends Student  {


        Grad(String name ,int id, int age,  String addr)
        {
```

```
            student_name = name ;
            student_ID = id;
            student_age = age;
            student_address= addr;

            System.out.println("Student_Name      :  " +student_name   );
            System.out.println("Student_ID        :  " +student_ID   );
            System.out.println("Student_Age       :  " +student_age   );
            System.out.println("Student_Address   :  " +student_address   );

        }
        String result = "";

        public boolean isPassed ( double grade )
        {
            student_grade = grade;
        System.out.println("Student_grad      :  " +student_grade   );
        if (student_grade>80)
        {
            return true;
        }
        else
            return false;

        }
}

public class inheritance_student_grade {

        static public void passed()
        {
            System.out.println("Result           :  Passed" );
        }
        static public void failed()
        {
            System.out.println("Result           :  Failed" );
        }

        public static void main(String[] NITHISH) {

            System.out.print("\t  Name           : Nithish G \n\t  Regno No.: 19BCS0012\n");
            System.out.print("\t  --------------------\n");
```

```
            System.out.println("\t\t Grad");
            System.out.print("\t\t------\n");

            Grad[] ob2 = new Grad[2];
            ob2[0] = new Grad("Nithish" ,12 , 18, "vellore");

            if(ob2[0].isPassed(88.8)) passed();

            else failed();
            System.out.println();

            ob2[1] = new Grad("Kumar" ,13 , 18,"chennai");
            if(ob2[1].isPassed(62.2)) {        passed(); }

            else            {failed();}
            System.out.print("-----------------------------------------------\n");

            System.out.println("\t      UnderGrad");
            System.out.print("\t      ---------\n");
            Undergrad[] ob1 = new Undergrad[2];


            ob1[0] = new Undergrad("Viknesh" ,14 , 17, "vellore");
            if(ob1[0].isPassed(72.8)) {passed();}

            else            {failed();}

            System.out.println();

            ob1[1] = new Undergrad("Ramesh" ,15 , 17, "vellore");

            if(ob1[1].isPassed(68.8)) { passed();  }

            else    {failed();}

        }
}
```

Output:

```
<terminated> inheritance_student_grade [Java Application] C:\Pr
            Name      : Nithish G
            Regno No.: 19BCS0012
            --------------------
                   Grad
                  ------
Student_Name       :  Nithish
Student_ID         :  12
Student_Age        :  18
Student_Address    :  vellore
Student_grad       :  88.8
Result             :  Passed

Student_Name       :  Kumar
Student_ID         :  13
Student_Age        :  18
Student_Address    :  chennai
Student_grad       :  62.2
Result             :  Failed
-----------------------------------------
                UnderGrad
                ---------
Student_Name       :  Viknesh
Student_ID         :  14
Student_Age        :  17
Student_Address    :  vellore
Student_grad       :  72.8
Result             :  Passed

Student_Name       :  Ramesh
Student_ID         :  15
Student_Age        :  17
Student_Name       :  Ramesh
Student_ID         :  15
Student_Age        :  17
Student_Address    :  vellore
Student_grad       :  68.8
Result             :  Failed
```

```
Lab ▸ src ▸ (default
79
80
81
82
83  ritance_student_gra
84
85  void main(String[]
86
87  t.print("\t  Name
88  t.print("\t  ------
89
90  t.println("\t\t Gra
91  t.print("\t\t------
92
93  2 = new Grad[2];
94  new Grad("Nithish"
95  Passed(88.8);
96  t.println();
97
98  new Grad("Kumar" ,1
99  Passed(62.2);
100 t.print("----------
101 t.println("\t
102 t.print("\t      --
103 [] ob1 = new Underg
104
105 new Undergrad("Vikn
106 Passed(72.8);
107
```

```
Problems  @ Javadoc  Decla
105 new Undergrad("Vikn
106 Passed(72.8);
107
```

```
Problems  @ Javadoc  Decla
inheritance_student_grade.java
```

5. Write a Java program which has Interface class for Stack Operations.
   (i) A Class that implements the Stack Interface and creates a fixed length Stack

Source Code:

```
import java.util.Scanner;
import java.util.NoSuchElementException;
```

```
interface Stack {

 void push(int data) throws Exception;

 int pop() throws Exception;

 int top() throws Exception;

 boolean isEmpty();

 int size();
}


class FixedArrayStack implements Stack {

 protected int[] stack;

 protected int top = -1;
 public static final int CAPACITY = 3;

 public FixedArrayStack() {
   stack = new int [FixedArrayStack.CAPACITY];
 }


 public void push(int data) throws Exception {
  if (size() == CAPACITY){
    throw new IndexOutOfBoundsException("StackOrverflowException.");
  } else {
    stack[++top] = data;
  }
 }

 public int top() throws Exception {
  if (isEmpty())
    throw new NoSuchElementException("Stack is empty.");
  return stack[top];
 }

 public int pop() throws Exception {
  int data;
  if (isEmpty())
    throw new NoSuchElementException("Stack is empty.");
```

```
    data = stack[top];


    stack[top--] = Integer.MIN_VALUE;
    return data;
  }

  public int size() {
    return (top + 1);
  }

  public boolean isEmpty() {
    return (top < 0);
  }


  public String toString() {
    String s = "[";
    if (size() > 0)
      s += stack[0];
    if (size() > 1)
      for (int i = 1; i <= size() - 1; i++) {
        s += ", " + stack[i];
      }
    return s + "]";
  }
}

public class Stack_interface_fixed_length {

  static Scanner scanner = new Scanner(System.in);



  public static void main(String[] args) {

          System.out.print("\t  Name         : Nithish G \n\t  Regno No.: 19BCS0012\n");
          System.out.print("\t  --------------------\n");

     Stack stack = new FixedArrayStack();
     System.out.println("      Stack created with size : " + FixedArrayStack.CAPACITY);
     System.out.println("      ----------------------------");

     while(true){
    System.out.println("\n  1      2     3       4        5 ");
    System.out.print(" push() pop() top() iterate() exit(1)\n Choice : ");
```

```
int choice = scanner.nextInt();

try {

  switch(choice) {
 case 1:
    System.out.print(" Enter a number to push : ");
    stack.push(scanner.nextInt());
    System.out.print(" ----------------------------------------");
    break;

 case 2:
    System.out.println(" popped element : " + stack.pop());
    System.out.print(" ----------------------------------------");
    break;

 case 3:
    System.out.println(" top element : " + stack.top());
    System.out.print(" ----------------------------------------");
    break;

 case 4:
     System.out.println(" elements are : " + stack.toString());
     System.out.print(" ----------------------------------------");
     break;

 case 5:
     System.out.print(" exit the sytem: ");
     System.exit(1);
   break;

 default :
   System.out.println("invalid option.........");

 }

} catch (Exception e) {
 System.err.println(" error message : " + e.getMessage());
}
  }
 }
}
```

Output:

```
<terminated> Stack_interface_fixed_length [Java Application] C:\Prog

          Name      : Nithish G
          Regno No.: 19BCS0012
          --------------------
      Stack created with size : 3
      ----------------------------

   1       2       3        4         5
  push() pop() top() iterate() exit(1)
 Choice : 1
Enter a number to push : 12
-----------------------------------------
   1       2       3        4         5
  push() pop() top() iterate() exit(1)
 Choice : 1
Enter a number to push : 13
-----------------------------------------
   1       2       3        4         5
  push() pop() top() iterate() exit(1)
 Choice : 1
Enter a number to push : 14
-----------------------------------------
   1       2       3        4         5
  push() pop() top() iterate() exit(1)
 Choice : 1
Enter a number to push : 15
error message : StackOrverflowException.
-----------------------------------------
   1       2       3        4         5
  push() pop() top() iterate() exit(1)
 Choice : 4
elements are : [12, 13, 14]
-----------------------------------------
   1       2       3        4         5
  push() pop() top() iterate() exit(1)
 Choice : 3
top element : 14
-----------------------------------------
   1       2       3        4         5
  push() pop() top() iterate() exit(1)
 Choice : 2
popped element : 14
-----------------------------------------
   1       2       3        4         5
  push() pop() top() iterate() exit(1)
 Choice : 2
popped element : 13
-----------------------------------------
   1       2       3        4         5
  push() pop() top() iterate() exit(1)
 Choice : 2
popped element : 12
-----------------------------------------
```

```
(default packag
81
82
83
84 public static void
85
86     System.out.pri
87     System.out.pri
88
89   Stack stack = n
90   System.out.prin
91   System.out.prin
92
93   while(true){
94  System.out.printl
95  System.out.print(
96
97  int choice = scan
98
99  try {
100
101     switch(choice)
102   case 1:
103     System.out.pr
104     stack.push(sc
105     System.out.pr
106     break;
107
108   case 2:
109     System.out.pr
```

Problems | @ Javadoc | Decl

```
84 public static void
85
86     System.out.pri
87     System.out.pri
88
89   Stack stack = n
90   System.out.prin
91   System.out.prin
92
93   while(true){
94  System.out.printl
95  System.out.print(
96
97  int choice = scan
98
99  try {
100
101     switch(choice)
102   case 1:
103     System.out.pr
104     stack.push(sc
```

```
-----------------------------------------
   1       2       3        4         5
  push()  pop()  top()  iterate()  exit(1)
 Choice : 2
error message : Stack is empty.
-----------------------------------------
   1       2       3        4         5
  push()  pop()  top()  iterate()  exit(1)
 Choice : 5
exit the Stack :
```

```
104        stack.push(sc
105        System.out.pr
106        break;
107
108    case 2:
109        System.out.pr
```

Problems | @ Javadoc | Decl

(ii) A Class that implements the Stack Interface and creates a Dynamic length Stack.

Source code

```
import java.util.Scanner;
import java.util.NoSuchElementException;

 interface dynamicStack {

  void push(int data) throws Exception;

  int pop() throws Exception;

  int top() throws Exception;

  boolean isEmpty();

  int size();
}


 class dynamicArrayStack implements Stack {

  protected int capacity;

  protected int[] stack;

  protected int top = -1;

   public dynamicArrayStack(int length) {
    this.capacity = length;
    stack = new int[this.capacity];
   }
```

```
public void push(int data) throws Exception {
  if (size() == capacity){
    throw new IndexOutOfBoundsException("StackOrverflowException.");
  } else {
    stack[++top] = data;
  }
}

public int top() throws Exception {
  if (isEmpty())
    throw new NoSuchElementException("Stack is empty.");
  return stack[top];
}

public int pop() throws Exception {
  int data;
  if (isEmpty())
    throw new NoSuchElementException("Stack is empty.");
  data = stack[top];

     stack[top--] = Integer.MIN_VALUE;
  return data;
}

public int size() {
  return (top + 1);
}

public boolean isEmpty() {
  return (top < 0);
}
```

```
  public String toString() {
    String s = "[";
    if (size() > 0)
      s += stack[0];
    if (size() > 1)
      for (int i = 1; i <= size() - 1; i++) {
        s += ", " + stack[i];
      }
    return s + "]";
  }
}

public class Stack_interface_dynamic_length {
    static Scanner read = new Scanner(System.in);

    public static void main(String[] args) {

            System.out.print("\t  Name         : Nithish G \n\t  Regno No.: 19BCS0012\n");
            System.out.print("\t  --------------------\n");

           System.out.print("Enter the Size of Stack : ");
           int size = read.nextInt();
     Stack stack = new dynamicArrayStack(size);
     System.out.println("      Stack created with size : " + FixedArrayStack.CAPACITY);
     System.out.println("      ----------------------------");

     while(true){
    System.out.println("\n  1      2     3       4        5 ");
    System.out.print(" push() pop() top() iterate() exit(1)\n Choice : ");
```

```
int choice = read.nextInt();

try {

  switch(choice) {
 case 1:
    System.out.print(" Enter a number to push : ");
    stack.push(read.nextInt());
    System.out.print(" ----------------------------------------");
    break;

 case 2:
    System.out.println(" popped element : " + stack.pop());
    System.out.print(" ----------------------------------------");
    break;

 case 3:
    System.out.println(" top element : " + stack.top());
    System.out.print(" ----------------------------------------");
    break;

 case 4:
     System.out.println(" elements are : " + stack.toString());
     System.out.print(" ----------------------------------------");
     break;

 case 5:
     System.out.print(" exit the sytem: ");
     System.exit(1);
  break;
```

```
    default :
      System.out.println("invalid option.....");


    }


  } catch (Exception e) {
    System.err.println(" error message : " + e.getMessage());
  }
    }
  }
}
```

Output

```
<terminated> Stack_interface_dynamic_length [Java Application] C:\F
            Name      : Nithish G
            Regno No.: 19BCS0012
            --------------------
       Enter the Size of Stack : 2
       Stack created with size : 2
       ----------------------------

  1       2       3        4         5
 push() pop() top() iterate() exit(1)
 Choice : 1
 Enter a number to push : 12
 ----------------------------------------
  1       2       3        4         5
 push() pop() top() iterate() exit(1)
 Choice : 1
 Enter a number to push : 13
 ----------------------------------------
  1       2       3        4         5
 push() pop() top() iterate() exit(1)
 Choice : 1
 Enter a number to push : 14
 Error message : StackOrverflowException.

  1       2       3        4         5
 push() pop() top() iterate() exit(1)
 Choice : 4
 elements are : [12, 13]
 ----------------------------------------
  1       2       3        4         5
 push() pop() top() iterate() exit(1)
 Choice : 3
```

```
Lab ▸ src ▸ (defa
82
83
84
85 blic static void
86
87   System.out.prin
88   System.out.prin
89
90  System.out.print
91  int size = read.
92  Stack stack = ne
93  System.out.print
94  System.out.print
95
96  while(true){
97 ystem.out.println
98 ystem.out.print("
99
100
101 nt choice = read.
102
103 ry {
104
105   switch(choice)
106 case 1:
107    System.out.pri
108    stack.push(rea
109    System.out.pri
110    break;
Problems  @ Javadoc  De
```

```
top element : 13
-----------------------------------------
 1       2       3        4         5
push() pop() top() iterate() exit(1)
Choice : 2
popped element : 13
-----------------------------------------
 1       2       3        4         5
push() pop() top() iterate() exit(1)
Choice : 2
popped element : 12
-----------------------------------------
 1       2       3        4         5
push() pop() top() iterate() exit(1)
Choice : 2
Error message : Stack is empty.

 1       2       3        4         5
push() pop() top() iterate() exit(1)
Choice : 5
Exit the Stack....
```

```
 System.out.prin
 System.out.prin

 while(true){
ystem.out.printl
ystem.out.print(


nt choice = read

ry {

  switch(choice)
case 1:
   System.out.pr
   stack.push(re
   System.out.pr
   break;
```

Problems | @ Javadoc | D

6. Write a Java program using Synchronized Threads, which demonstrates Producer Consumer concepts.

Source Code:

```
class Shop1
        {

        int materials;
        boolean spaceavailable = false;

        public synchronized void put(int materials)
        {

        while (spaceavailable)
        {
        try{ wait(); } catch( Exception e ) {}
        }

        System.out.println("Produced value --> put method : " + materials);
        this.materials = materials;
```

```
spaceavailable = true;
notify();
}
public synchronized void get()
{
while (!spaceavailable)
{
try { wait (); } catch( Exception e ) {}
}
System.out.println("Consumer value --> put method : " + materials);
System.out.println("");
spaceavailable = false;
notify();
}
}

class Producer1 implements Runnable
{

Shop1 shop;

public  Producer1(Shop1 shop)
{
this.shop =shop;
Thread t = new Thread(this, "Producer");
t.start();
}

public void run()
   {
int i =0;
while(true)
        {
shop.put(i++);
try { Thread.sleep(1000) ; } catch(Exception e) {}
        }
   }
}
```

```
class Consumer1 implements Runnable
{
 Shop1 shop;

public Consumer1(Shop1 shop)
{
this.shop = shop;
Thread t = new Thread (this, "Consumer1");
t.start();
}
public void run()
{
while ( true)
 {
   shop.get();
         try{ Thread.sleep(1000);} catch( Exception e) {}
      }
   }
         }
public class Producer_Consumer
{
    public static void main(String[] Nithish)
    {
     System.out.print("\tName   : Nithish G \n\tReg No. : 19BCS0012\n\n");
    Shop1 shop = new Shop1();
         new Producer1(shop);
         new Consumer1(shop);
    }
}
```

Output:

```
Package Explorer   Console
<terminated> Producer_Consumer [Java Application] C:\P

        Name     : Nithish G
        Reg No.  : 19BCS0012
-------------------------------------

Produced value --> put method : 0
Consumer value --> put method : 0

Produced value --> put method : 1
Consumer value --> put method : 1

Produced value --> put method : 2
Consumer value --> put method : 2

Produced value --> put method : 3
Consumer value --> put method : 3

Produced value --> put method : 4
Consumer value --> put method : 4

Produced value --> put method : 5
Consumer value --> put method : 5

Produced value --> put method : 6
Consumer value --> put method : 6

Produced value --> put method : 7
Consumer value --> put method : 7
```

```
Main.java   Producerconsumer2.ja
(default package)
64  p;
65  Thread (this, "Consumer1"
66
67
68  .()
69
70
71
72
73   Thread.sleep(1000);} cat
74
75
76
77
78
79
80  oducer_Consumer
81
82  .tic void main(String[] Ni
83
84
85  .out.print("\tName    : Ni
86  .out.print("-------------
87  p = new Shop1();
88  .cer1(shop);
89  mer1(shop);
90
91
92
```

------- Thank You Madam! -------