Name Nikhil Gupta

Roll No. 2005523

Batch CSE-06

Subject Web Tech

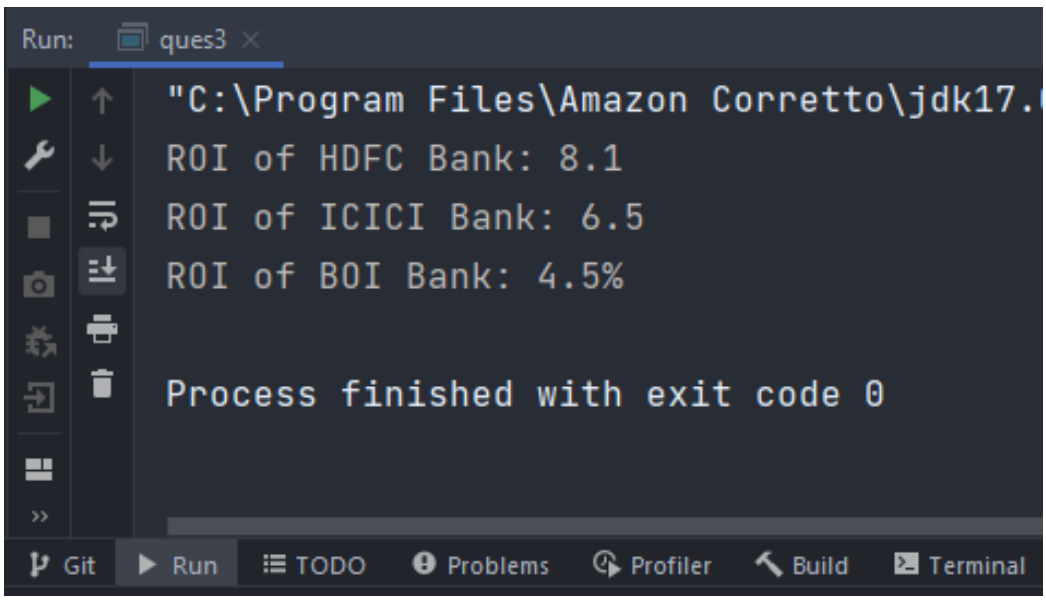
Assignment 9

Ques 2

```
class Bank {
    Double ROI;
    void findROI() {
        System.out.println("ROI:" + ROI);
    }
}

class HDFC extends Bank {
    HDFC() {
        ROI = 8.1;
    }
    void findROI() {
        System.out.println("ROI of HDFC Bank: " + ROI);
    }
}
```

```
class ICICI extends Bank {
    ICICI() {
        ROI = 6.5;
    }
    void findROI() {
        System.out.println("ROI of ICICI Bank: "+ROI);
    }
}

class BOI extends Bank {
    BOI() {
        ROI = 4.5;
    }
    void findROI() {
        System.out.println("ROI of BOI Bank: "+ROI);
    }
}

public class ques3 {
    public static void main(String[] args) {
        HDFC hdfc = new HDFC();
```

```
ICICI icici = new ICICI();
BOI boi = new BOI();
Bank ref;
ref = hdfc;
ref.findROI();
ref = icici;
ref.findROI();
ref = boi;
ref.findROI();
}
}
```

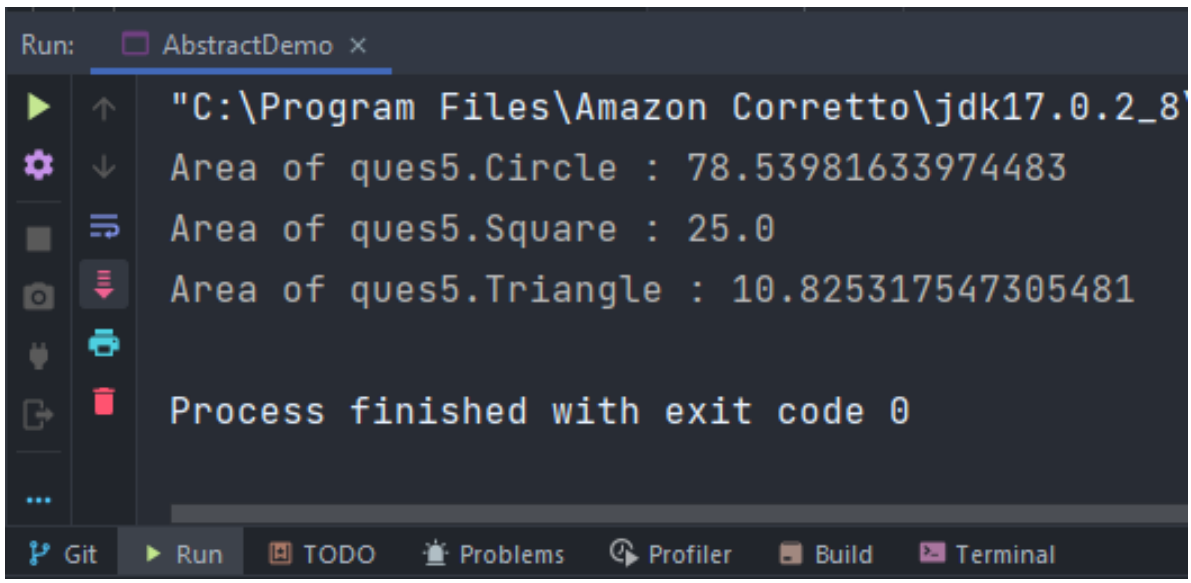


Ques 1

```
public class AbstractDemo
{
    public static void main(String[] args)
    {
        Shape shape;
        Circle crcl = new Circle();
        shape = crcl;
        shape.setValues(5);
        System.out.println("Area of Circle: " +
                                shape.getArea());
        Square sq = new Square();
        shape = sq;
        shape.setValues(5);
        System.out.println("Area of Square: " +
                                shape.getArea());
        Triangle tri = new Triangle();
        shape = tri;
        shape.setValues(5);
        System.out.println("Area of Triangle: " + shape.
                                getArea());
    }
}
```

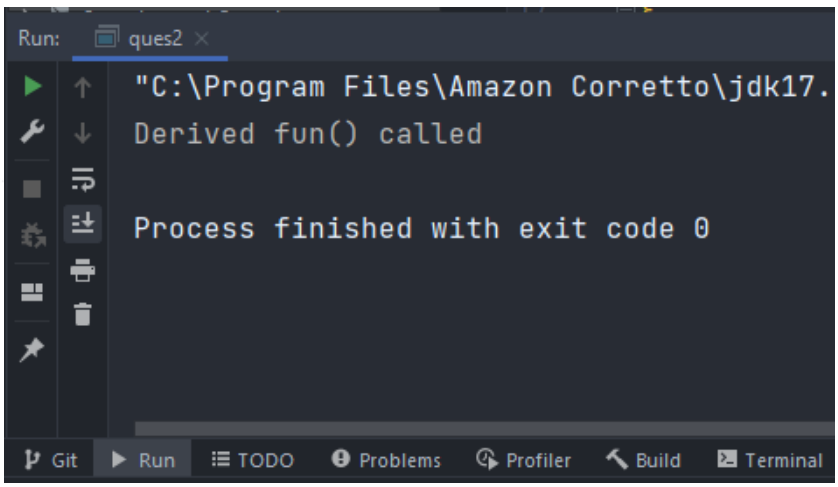
```
public abstract class Shape
 {
    private double radius;
    public void setValues(double radius)
    {
      this.radius = radius;
    }
    public double getRadius()
     {
       return radius;
     }
    public abstract double getArea();
 }
public class Square extends Shape
 {
   public double getArea()
    {
      return getRadius() * getRadius();
    }
}
```

```
public class Triangle extends Shape {

    public double getArea()
    {

    return Math.sqrt(3) * 0.25 * getRadius() *
                                 getRadius();

    }

}

public class Circle extends Shape
 {

    public double getArea()
     {

    return Math.PI *(getRadius() * getRadius());

    }

}
```



Ques 2

```
abstract class Base {
    abstract void fun();
}

class derived extends Base {
    void fun()
    {
      System.out.println("Derived fun() called");
    }
}

class Main {
    public static void main(String [] args)
    {
        base b = new Derived();
        b.fun();
    }
}
```

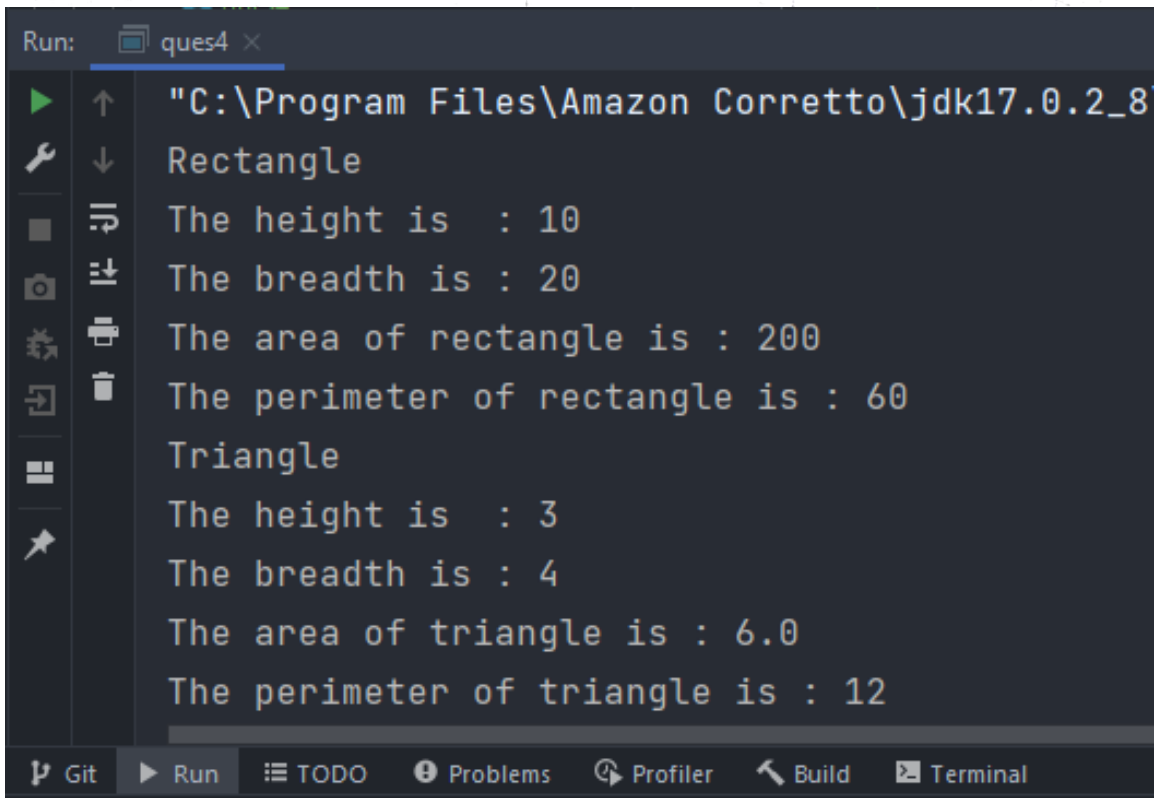


Ques 4

```
class Data
{
    int height, breadth;
    void getData(int h, int b)
    {
        height = h;
        breadth = b;
    }
    void putData()
    {
      System.out.println("Height: " + height +
                         "breadth: " + breadth);
    }
}

class Rectangle extends Data
{
    void areaRectangle()
    {
      System.out.println("Area of Rectangle" + (height *
                                                 breadth));
    }
}
```

```
class Triangle extends Data
{
    void areaofTriangle()
    {
      System.out.println("Area of Triangle: " +
                         (0.5 * height * breadth));
    }
    void perimeterofTriangle()
    {
      int s =(int)Math.sqrt(Math.pow(height,2)+
          Math.pow(breadth, 2));
      System.out.println("Perimeter of Triangle:" +
                         (height + breadth +s));
    }
public class geees4
  {
    public static void main (String[] args)
    {
      Rectangle obj1 = new Rectangle();
      Triangle obj2 = new Triangle();
      obj.getData(10, 20);
```

```
obj2.getData(3, 4);
System.out.println("Rectangle");
obj1.putData();
obj1.areaofRectangle();
obj1.area
obj1.perimeterofRectangle();
System.out.println("Triangle");
obj2.putData();
obj2.areaofTriangle();
obj2.perimeterofTriangle();
}
}
```



Ques 5

~~class stack()~~

```
class stack {
    static final int MAX = 10;
    int top;
    int[] a = new int[MAX];

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

  Stack()
   {
      top = -1;
   }
 boolean push(int x)
  {
     if(top >= (Max-1)) {
         System.out.println("overflow");
         return false;
 }
```

```
        else {
            a[++top] = x;
            System.out.println(x + "pushed into Stack");
            return true;
        }
    }
}

int pop()
{
    if(top<0) {
        System.out.println("Underflow");
        return 0;
    }
    else {
        int x = a[top--];
        return x;
    }
}

int display int top()
        {
            if(top<0) {
```

```
            System.out.println ("Underflow");
            return 0;
        }
        else {
            int x = a[top];
            return x;
        }
    }
    void display () {
        for(int i = top; i >=1; i--)
        {
            System.out.println (" "+ a[i]);
        }
    }
}

class ques5 {
    public static void main(String[] args)
    {
        Stack s = new Stack();
        s.push(10);
        s.push (15);
```

```
s.push(18);
s.push(20);
s.push(118);
System.out.println(s.pop() + "Popped out from
                                Stack");
System.out.println("Top Element is: " +
                                s.Top());
System.out.println("Elements in Stack: ");
s.display();
```