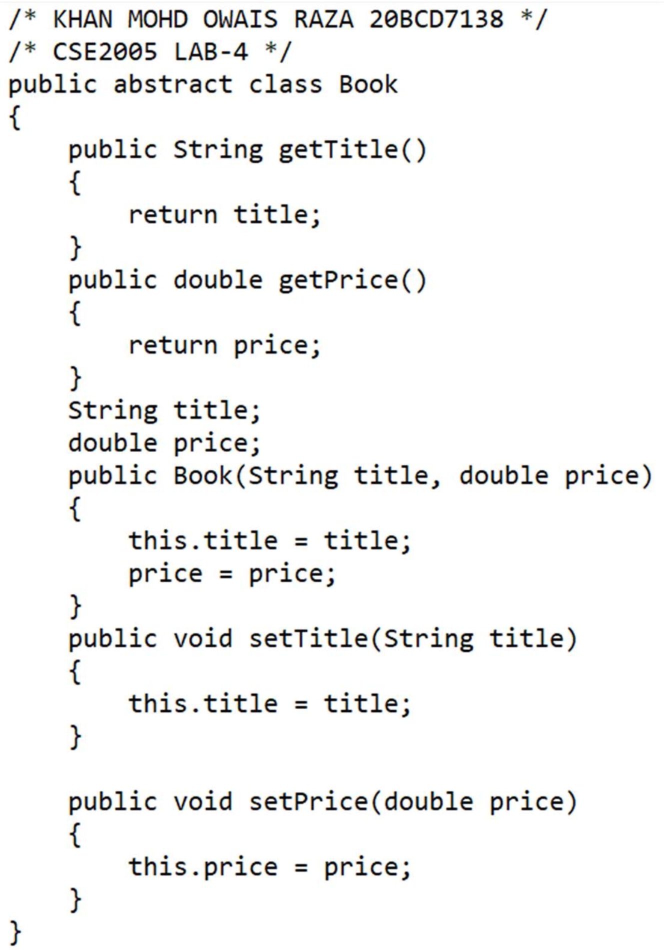
CSE2005 (OOP in Java) Lab-4 02/12/2021

KHAN MOHD. OWAIS RAZA
20BCD7138

Q.1]

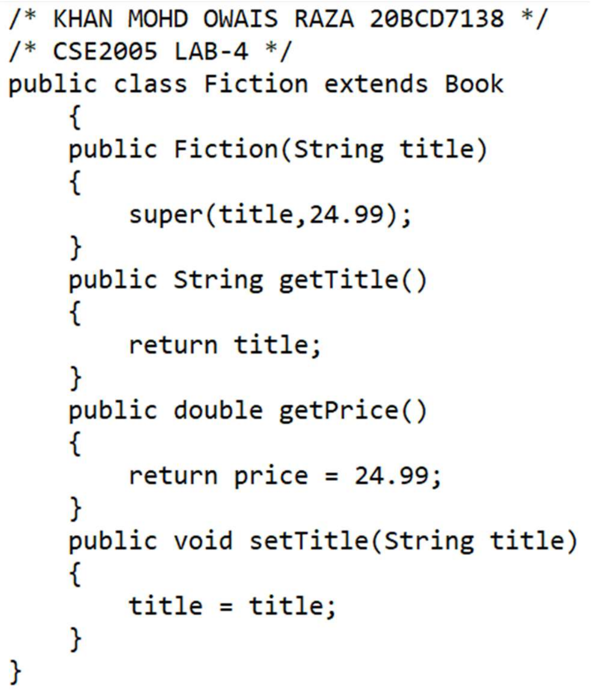
a) Create an abstract class named Book. Include a String field for the book’s title and a double field for the book’s price. Within the class, include a constructor that requires the book title, and add two get methods—one that returns the title and one that returns the price. Include an abstract method named setPrice(). Create two child classes of Book: Fiction and NonFiction. Each must include a setPrice() method that sets the price for all Fiction Books to $24.99 and for all NonFiction Books to $37.99. Write a constructor for each subclass, and include a call to setPrice() within each. Write an application demonstrating that you can create both a Fiction and a NonFiction Book, and display their fields. Save the files as Book.java, Fiction.java, NonFiction.java, and UseBook.java.

Book.java :–



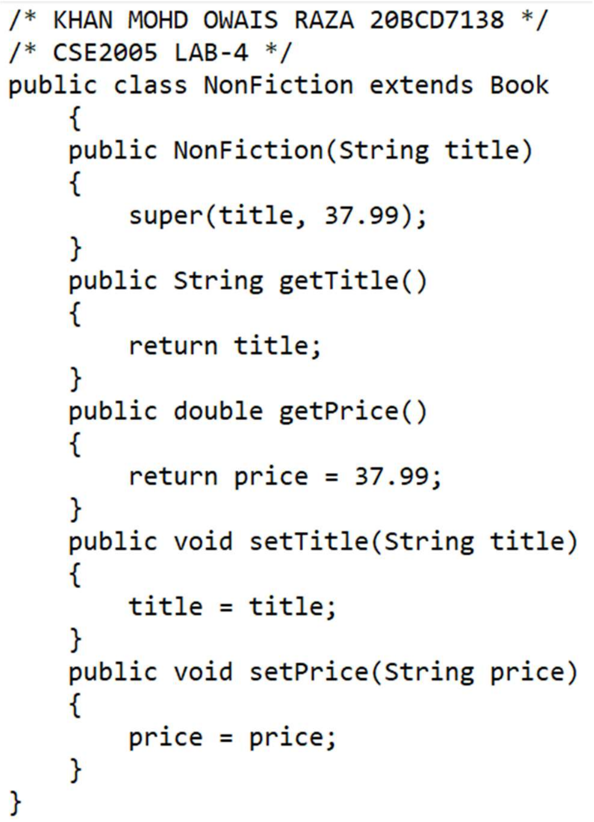
```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public abstract class Book
{
    public String getTitle()
    {
        return title;
    }
    public double getPrice()
    {
        return price;
    }
    String title;
    double price;
    public Book(String title, double price)
    {
        this.title = title;
        price = price;
    }
    public void setTitle(String title)
    {
        this.title = title;
    }

    public void setPrice(double price)
    {
        this.price = price;
    }
}
```

Fiction.java :–



```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public class Fiction extends Book
    {
    public Fiction(String title)
    {
        super(title,24.99);
    }
    public String getTitle()
    {
        return title;
    }
    public double getPrice()
    {
        return price = 24.99;
    }
    public void setTitle(String title)
    {
        title = title;
    }
}
```

NonFiction.java :–

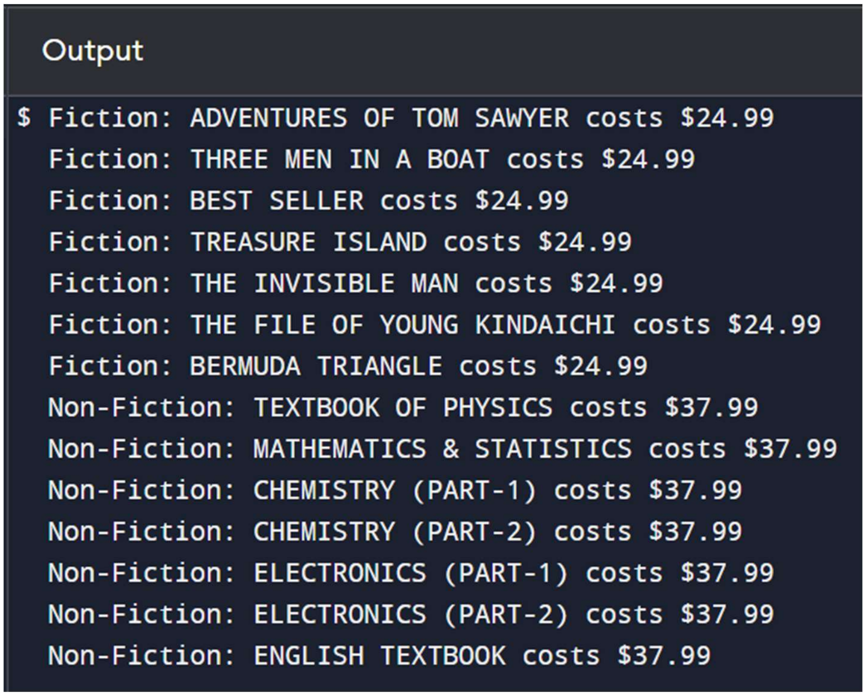


```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public class NonFiction extends Book
    {
    public NonFiction(String title)
    {
        super(title, 37.99);
    }
    public String getTitle()
    {
        return title;
    }
    public double getPrice()
    {
        return price = 37.99;
    }
    public void setTitle(String title)
    {
        title = title;
    }
    public void setPrice(String price)
    {
        price = price;
    }
}
```

UseBook.java :–

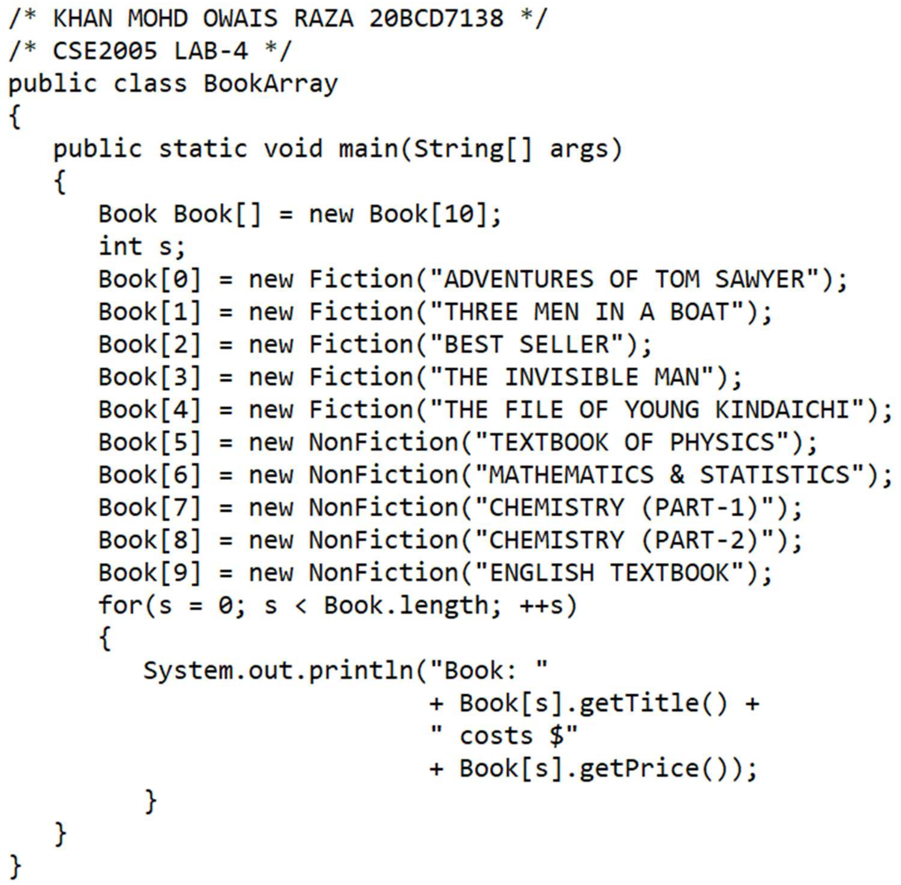
```java
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public class UseBook
{
    public static void main(String[] args)
    {
        Fiction aNovel = new Fiction("ADVENTURES OF TOM SAWYER");
        Fiction aNovel = new Fiction("THREE MEN IN A BOAT");
        Fiction aNovel = new Fiction("BEST SELLER");
        Fiction aNovel = new Fiction("TREASURE ISLAND");
        Fiction aNovel = new Fiction("THE INVISIBLE MAN");
        Fiction aNovel = new Fiction("THE FILE OF YOUNG KINDAICHI");
        Fiction aNovel = new Fiction("BERMUDA TRIANGLE");
        NonFiction aTextbook = new NonFiction("TEXTBOOK OF PHYSICS");
        NonFiction aTextbook = new NonFiction("MATHEMATICS & STATISTICS");
        NonFiction aTextbook = new NonFiction("CHEMISTRY (PART-1)");
        NonFiction aTextbook = new NonFiction("CHEMISTRY (PART-2)");
        NonFiction aTextbook = new NonFiction("ELECTRONICS (PART-1)");
        NonFiction aTextbook = new NonFiction("ELECTRONICS (PART-2)");
        NonFiction aTextbook = new NonFiction("ENGLISH TEXTBOOK");
        System.out.println("Fiction: "
                           + aNovel.getTitle() +
                           " costs $"
                           + aNovel.getPrice());
        System.out.println("Non-Fiction: "
                           + aTextbook.getTitle() +
                           " costs $"
                           + aTextbook.getPrice());
    }
}
```

Output :–



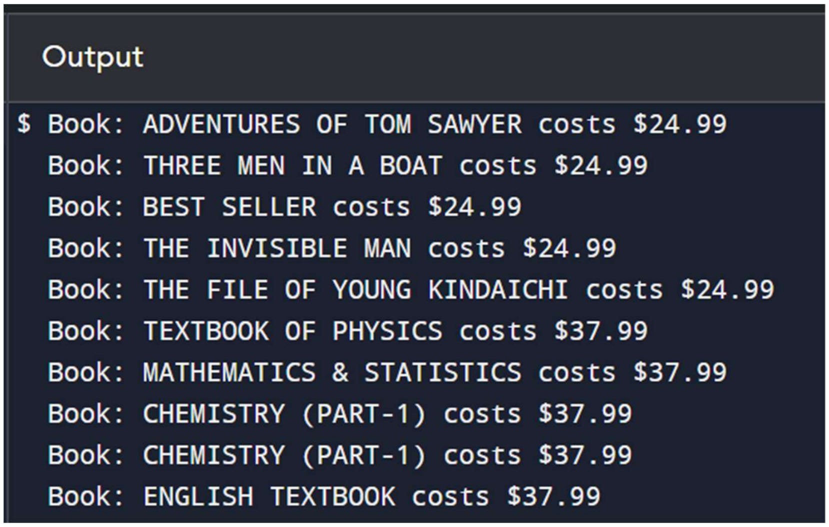
```
$ Fiction: ADVENTURES OF TOM SAWYER costs $24.99
  Fiction: THREE MEN IN A BOAT costs $24.99
  Fiction: BEST SELLER costs $24.99
  Fiction: TREASURE ISLAND costs $24.99
  Fiction: THE INVISIBLE MAN costs $24.99
  Fiction: THE FILE OF YOUNG KINDAICHI costs $24.99
  Fiction: BERMUDA TRIANGLE costs $24.99
  Non-Fiction: TEXTBOOK OF PHYSICS costs $37.99
  Non-Fiction: MATHEMATICS & STATISTICS costs $37.99
  Non-Fiction: CHEMISTRY (PART-1) costs $37.99
  Non-Fiction: CHEMISTRY (PART-2) costs $37.99
  Non-Fiction: ELECTRONICS (PART-1) costs $37.99
  Non-Fiction: ELECTRONICS (PART-2) costs $37.99
  Non-Fiction: ENGLISH TEXTBOOK costs $37.99
```

b) Write an application named BookArray in which you create an array that holds 10 Books, some Fiction and some NonFiction. Using a for loop, display details about all 10 books. Save the file as BookArray.java.



```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public class BookArray
{
   public static void main(String[] args)
   {
      Book Book[] = new Book[10];
      int s;
      Book[0] = new Fiction("ADVENTURES OF TOM SAWYER");
      Book[1] = new Fiction("THREE MEN IN A BOAT");
      Book[2] = new Fiction("BEST SELLER");
      Book[3] = new Fiction("THE INVISIBLE MAN");
      Book[4] = new Fiction("THE FILE OF YOUNG KINDAICHI");
      Book[5] = new NonFiction("TEXTBOOK OF PHYSICS");
      Book[6] = new NonFiction("MATHEMATICS & STATISTICS");
      Book[7] = new NonFiction("CHEMISTRY (PART-1)");
      Book[8] = new NonFiction("CHEMISTRY (PART-2)");
      Book[9] = new NonFiction("ENGLISH TEXTBOOK");
      for(s = 0; s < Book.length; ++s)
      {
         System.out.println("Book: "
                              + Book[s].getTitle() +
                              " costs $"
                              + Book[s].getPrice());
      }
   }
}
```

Output :–



```
$ Book: ADVENTURES OF TOM SAWYER costs $24.99
  Book: THREE MEN IN A BOAT costs $24.99
  Book: BEST SELLER costs $24.99
  Book: THE INVISIBLE MAN costs $24.99
  Book: THE FILE OF YOUNG KINDAICHI costs $24.99
  Book: TEXTBOOK OF PHYSICS costs $37.99
  Book: MATHEMATICS & STATISTICS costs $37.99
  Book: CHEMISTRY (PART-1) costs $37.99
  Book: CHEMISTRY (PART-1) costs $37.99
  Book: ENGLISH TEXTBOOK costs $37.99
```

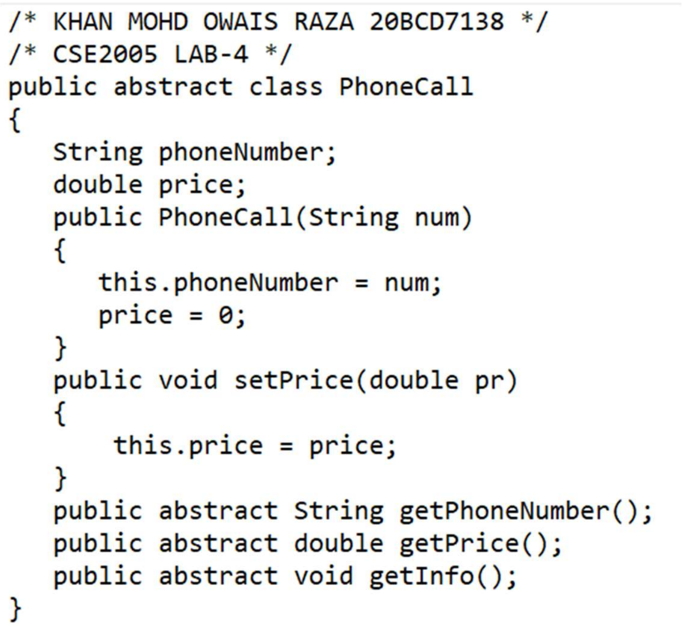
Q.2]

a) The Talk-A-Lot Cell Phone Company provides phone services for its customers. Create an abstract class named PhoneCall that includes a String field for a phone number and a double field for the price of the call. Also include a constructor that requires a phone number parameter and that sets the price to 0.0. Include a set method for the price. Also include three abstract get methods – one that returns the phone number, another that returns the price of the call, and a third that displays information about the call.

Create two child classes of PhoneCall: IncomingPhoneCall and OutgoingPhoneCall. The IncomingPhoneCall constructor passes its phone number parameter to its parent's constructor and sets the price of the call to 0.02. The method that displays the phone call information displays the phone number, the rate, and the price of the call (which is the same as the rate). The OutgoingPhoneCall class includes an additional field that holds the time of the call in minutes. The constructor requires both a phone number and the time. The price is 0.04 per minute, and the display method shows the details of the call, including the phone number, the rate per minute, the number of minutes, and the total price.

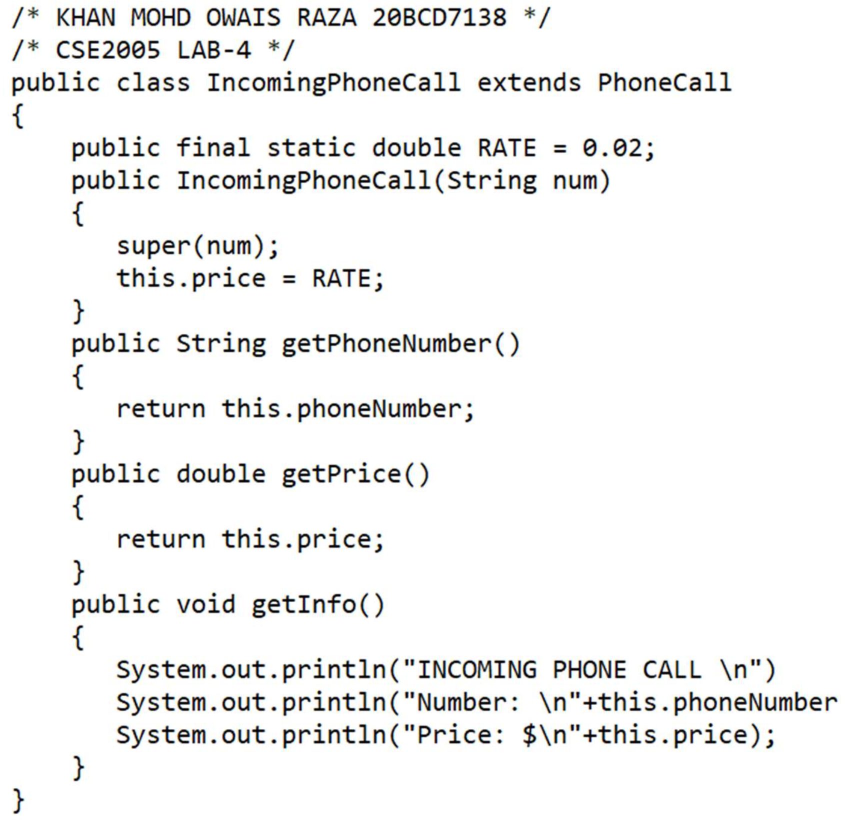
Write an application that demonstrates you can instantiate and display both IncomingPhoneCall and OutgoingPhoneCall objects. Save the files as PhoneCall.java, IncomingPhoneCall.java, OutgoingPhoneCall.java, and DemoPhoneCalls.java.

PhoneCall.java :–



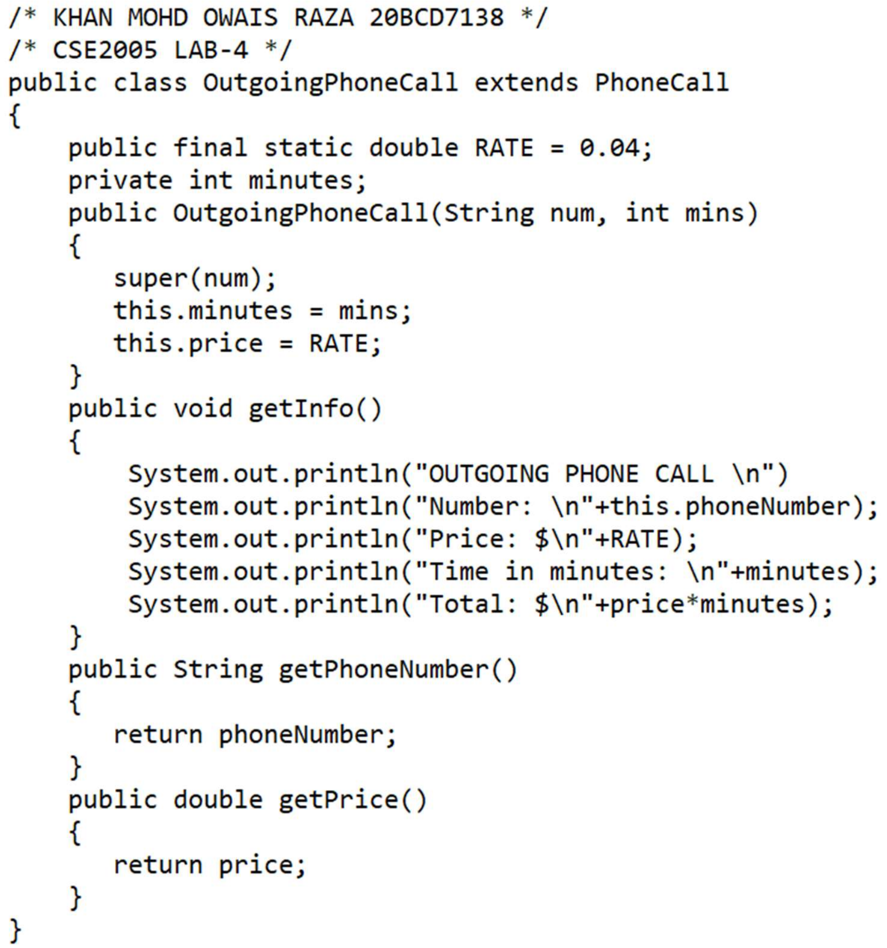
```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public abstract class PhoneCall
{
   String phoneNumber;
   double price;
   public PhoneCall(String num)
   {
      this.phoneNumber = num;
      price = 0;
   }
   public void setPrice(double pr)
   {
       this.price = price;
   }
   public abstract String getPhoneNumber();
   public abstract double getPrice();
   public abstract void getInfo();
}
```

IncomingPhoneCall.java :–



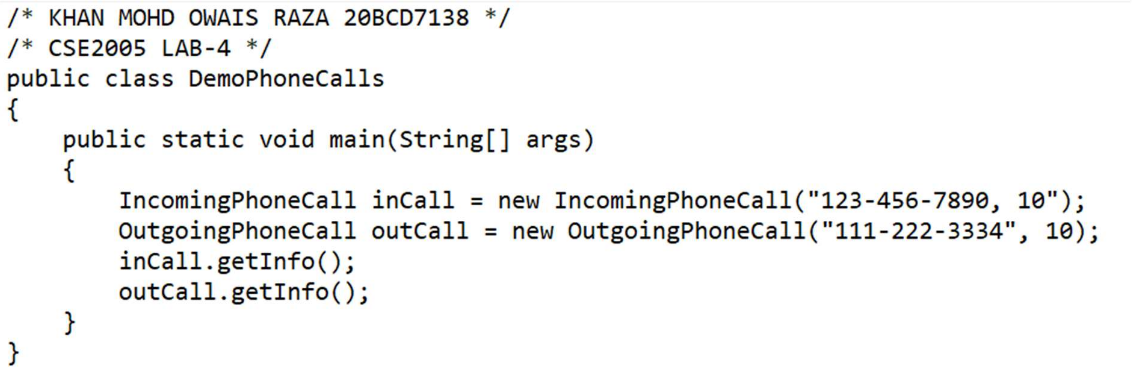
```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public class IncomingPhoneCall extends PhoneCall
{
    public final static double RATE = 0.02;
    public IncomingPhoneCall(String num)
    {
       super(num);
       this.price = RATE;
    }
    public String getPhoneNumber()
    {
       return this.phoneNumber;
    }
    public double getPrice()
    {
       return this.price;
    }
    public void getInfo()
    {
       System.out.println("INCOMING PHONE CALL \n")
       System.out.println("Number: \n"+this.phoneNumber
       System.out.println("Price: $\n"+this.price);
    }
}
```

OutgoingPhoneCall.java :–



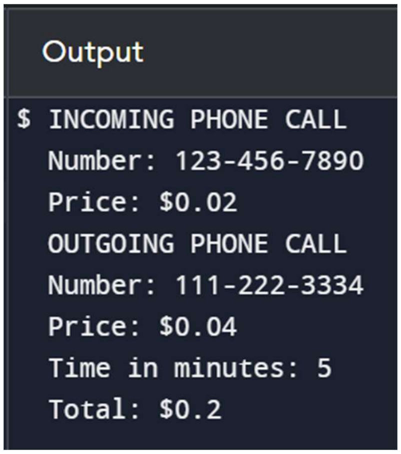
```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public class OutgoingPhoneCall extends PhoneCall
{
    public final static double RATE = 0.04;
    private int minutes;
    public OutgoingPhoneCall(String num, int mins)
    {
       super(num);
       this.minutes = mins;
       this.price = RATE;
    }
    public void getInfo()
    {
        System.out.println("OUTGOING PHONE CALL \n")
        System.out.println("Number: \n"+this.phoneNumber);
        System.out.println("Price: $\n"+RATE);
        System.out.println("Time in minutes: \n"+minutes);
        System.out.println("Total: $\n"+price*minutes);
    }
    public String getPhoneNumber()
    {
       return phoneNumber;
    }
    public double getPrice()
    {
       return price;
    }
}
```

DemoPhoneCalls.java :–



```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public class DemoPhoneCalls
{
    public static void main(String[] args)
    {
        IncomingPhoneCall inCall = new IncomingPhoneCall("123-456-7890, 10");
        OutgoingPhoneCall outCall = new OutgoingPhoneCall("111-222-3334", 10);
        inCall.getInfo();
        outCall.getInfo();
    }
}
```

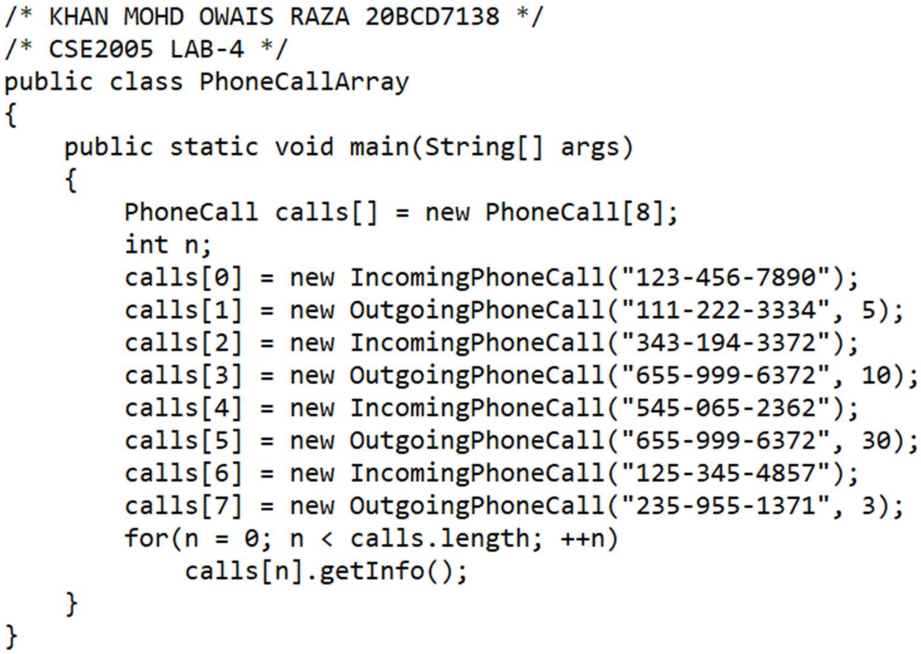
Output :–



```
$ INCOMING PHONE CALL
  Number: 123-456-7890
  Price: $0.02
  OUTGOING PHONE CALL
  Number: 111-222-3334
  Price: $0.04
  Time in minutes: 5
  Total: $0.2
```

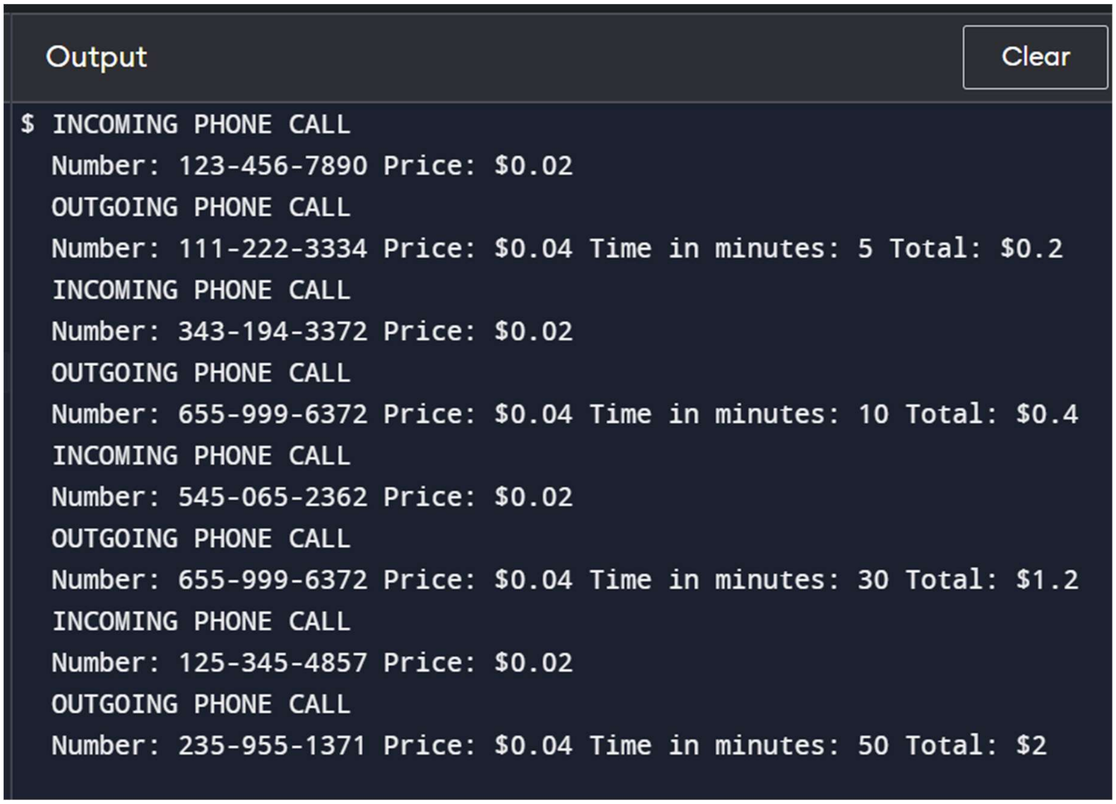
b) Write an application in which you assign data to a mix of eight IncomingPhoneCall and OutgoingPhoneCall objects into an array. Use a for loop to display the data. Save the file as PhoneCallArray.java.

PhoneCallArray.java :–



```java
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public class PhoneCallArray
{
    public static void main(String[] args)
    {
        PhoneCall calls[] = new PhoneCall[8];
        int n;
        calls[0] = new IncomingPhoneCall("123-456-7890");
        calls[1] = new OutgoingPhoneCall("111-222-3334", 5);
        calls[2] = new IncomingPhoneCall("343-194-3372");
        calls[3] = new OutgoingPhoneCall("655-999-6372", 10);
        calls[4] = new IncomingPhoneCall("545-065-2362");
        calls[5] = new OutgoingPhoneCall("655-999-6372", 30);
        calls[6] = new IncomingPhoneCall("125-345-4857");
        calls[7] = new OutgoingPhoneCall("235-955-1371", 3);
        for(n = 0; n < calls.length; ++n)
            calls[n].getInfo();
    }
}
```

Output :–

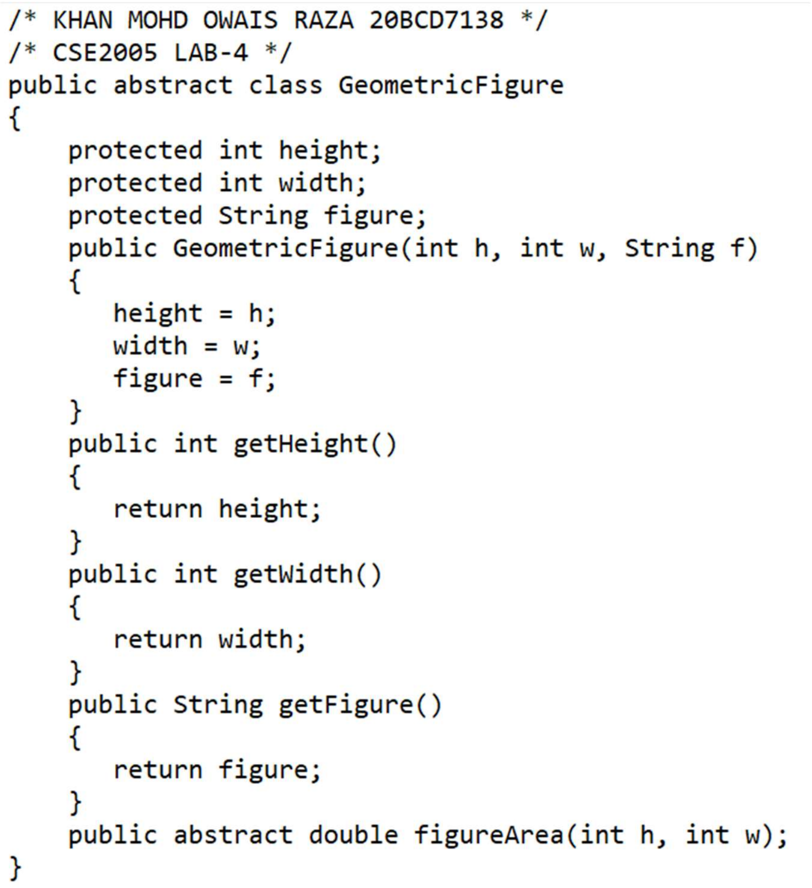


```
$ INCOMING PHONE CALL
  Number: 123-456-7890 Price: $0.02
  OUTGOING PHONE CALL
  Number: 111-222-3334 Price: $0.04 Time in minutes: 5 Total: $0.2
  INCOMING PHONE CALL
  Number: 343-194-3372 Price: $0.02
  OUTGOING PHONE CALL
  Number: 655-999-6372 Price: $0.04 Time in minutes: 10 Total: $0.4
  INCOMING PHONE CALL
  Number: 545-065-2362 Price: $0.02
  OUTGOING PHONE CALL
  Number: 655-999-6372 Price: $0.04 Time in minutes: 30 Total: $1.2
  INCOMING PHONE CALL
  Number: 125-345-4857 Price: $0.02
  OUTGOING PHONE CALL
  Number: 235-955-1371 Price: $0.04 Time in minutes: 50 Total: $2
```

Q.3] Create an abstract class called GeometricFigure. Each figure includes a height, a width, a figure type, and an area. Include an abstract method to determine the area of the figure. Create two subclasses called Square and Triangle.

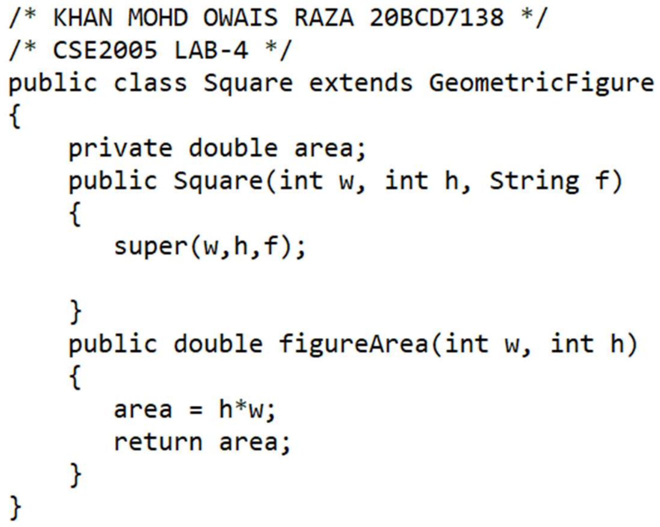
Create an application that demonstrates creating objects of both subclasses, and store them in an array. Save the files as GeometricFigure.java, Square.java, Triangle.java, and UseGeometric.java.

GeometricFigure.java :–



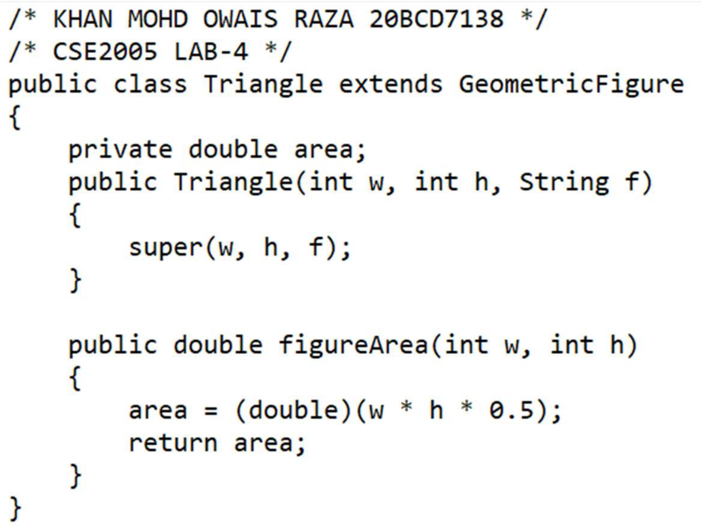
```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public abstract class GeometricFigure
{
    protected int height;
    protected int width;
    protected String figure;
    public GeometricFigure(int h, int w, String f)
    {
       height = h;
       width = w;
       figure = f;
    }
    public int getHeight()
    {
       return height;
    }
    public int getWidth()
    {
       return width;
    }
    public String getFigure()
    {
       return figure;
    }
    public abstract double figureArea(int h, int w);
}
```

Square.java :–



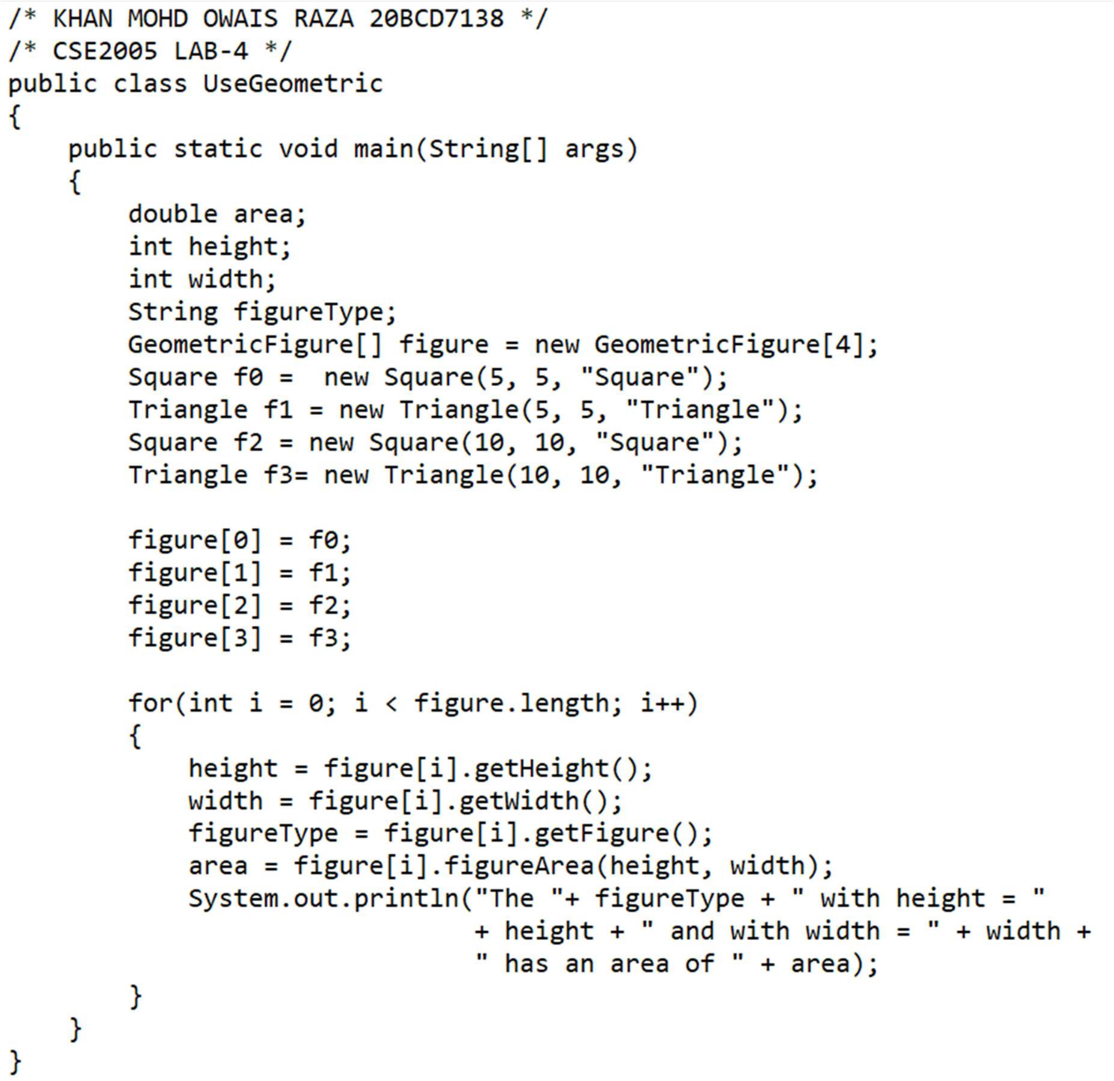
```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public class Square extends GeometricFigure
{
    private double area;
    public Square(int w, int h, String f)
    {
       super(w,h,f);

    }
    public double figureArea(int w, int h)
    {
       area = h*w;
       return area;
    }
}
```

Triangle.java :–



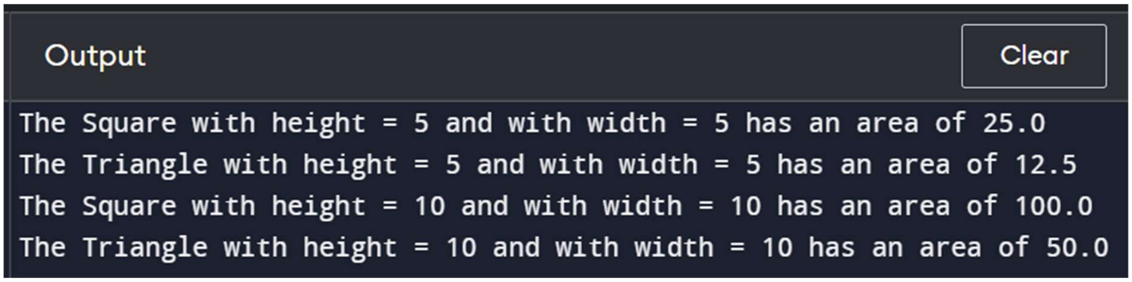
```java
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public class Triangle extends GeometricFigure
{
    private double area;
    public Triangle(int w, int h, String f)
    {
        super(w, h, f);
    }

    public double figureArea(int w, int h)
    {
        area = (double)(w * h * 0.5);
        return area;
    }
}
```

UseGeometric.java :–



```java
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-4 */
public class UseGeometric
{
    public static void main(String[] args)
    {
        double area;
        int height;
        int width;
        String figureType;
        GeometricFigure[] figure = new GeometricFigure[4];
        Square f0 =  new Square(5, 5, "Square");
        Triangle f1 = new Triangle(5, 5, "Triangle");
        Square f2 = new Square(10, 10, "Square");
        Triangle f3= new Triangle(10, 10, "Triangle");

        figure[0] = f0;
        figure[1] = f1;
        figure[2] = f2;
        figure[3] = f3;

        for(int i = 0; i < figure.length; i++)
        {
            height = figure[i].getHeight();
            width = figure[i].getWidth();
            figureType = figure[i].getFigure();
            area = figure[i].figureArea(height, width);
            System.out.println("The "+ figureType + " with height = "
                               + height + " and with width = " + width +
                               " has an area of " + area);
        }
    }
}
```

Output :–



```
The Square with height = 5 and with width = 5 has an area of 25.0
The Triangle with height = 5 and with width = 5 has an area of 12.5
The Square with height = 10 and with width = 10 has an area of 100.0
The Triangle with height = 10 and with width = 10 has an area of 50.0
```

----------------x----------------