1. Write exactly what this program will print after the main is executed.

public interface Shape {  
 int getWidth ();  
 int getLength ();  
 int getArea ();  
}

public abstract class AbstractShape implements Shape {  
 private int width;  
 private int length;  
  
 public int getWidth() {   
 return width;  
 }  
   
 public int getLength() {  
 return length;  
 }  
   
 public void setWidth(int val) {  
 width = val;  
 }  
   
 public void setLength(int val) {  
 length = val;  
 }  
   
 public AbstractShape(int width, int length) {  
 setWidth(width);  
 setLength(length);  
 }  
   
}

public class Rectangle extends AbstractShape{  
  
 public Rectangle(int width, int length) {  
 super(width, length);  
 }  
   
 public int getArea() {  
 return getLength()\*getWidth();  
 }  
   
 public String toString() {  
 return String.format("Rectangle: width: %d, length: %d, area: %d", getWidth(), getLength(), getArea());  
 }  
}

public class Triangle extends AbstractShape {  
 private double angle;  
   
 public int getArea() {  
 return (getWidth()\*getLength())/2;  
 }  
   
 public double getAngle() {   
 return angle;  
 }  
  
 public void setAngle(double val) {  
 angle = val;  
 }  
   
 public Triangle(int base, int height) {  
 this(base, height, 90.0);  
 }  
   
 public Triangle(int base, int height, double angle) {  
 super(base, height);  
 setAngle(angle);  
 }  
   
 public String toString() {  
 return String.format("Triangle: base: %d, height: %d, angle: %.2f, area: %d",   
 getWidth(), getLength(), getAngle(), getArea());  
 }  
   
}

public class Square extends Rectangle {  
 public Square(int side) {  
 this(side, side);  
 }  
   
 private Square(int width, int length) {  
 super(width, length);  
 }  
}

public class Main24 {  
 public static void main(String[] args) {  
 List<Object> shapes = new ArrayList<>();   
 shapes.add(new Rectangle(10, 2));  
 shapes.add(new Square(5));  
 shapes.add(new Triangle(5, 2));  
 shapes.add(new Triangle(5, 2, 23.33333));  
 shapes.add(new Rectangle(20, 2));  
 for(Object obj : shapes) System.out.println(obj);   
 }  
  
}

1. Given the following code, what is printed?  
     
   private static int doSomething(int[] arr) {  
     int val = 0;  
     for(int i = 0; i < arr.length; i+=1) {  
        val += arr[i];  
    }  
    return val + arr.length;  
   }

public static void main(String[] args) {  
 int[] arr = new int[10];  
 arr[0] = 10;  
 arr[1] = 2;  
 arr[0] = 1;  
 arr[2] = 20;  
 arr[3] = 4;  
 arr[8] = 6;  
 System.out.println(doSomething(arr));  
}

1. Given the following code, what type of sort does it perform?

void sort(int arr[]) {

int n = arr.length;

for (int i = 0; i < n-1; i++) {

for (int j = 0; j < n-i-1; j++) {

if (arr[j] > arr[j+1]) {

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

Hint: come up with an array, and quickly run/write down each pass to see what it is doing.

1. Given the following output, write a code that generates this output.

Print 1: [Anna, Matt]

Print 2: [Anna]

Print 3: [Anna, James]

Print 4: [Maria, Anna, Denny, James]

1. What is printed?

    public static boolean stringCheck(String s) {

        if (s.length() <= 1)

            return true;

        if (s.charAt(0) == s.charAt(s.length() - 1))

            return stringCheck(s.substring(1, s.length() - 1));

        return false;

    }

    public static void main(String[] args) {

        System.out.println(stringCheck("anna"));

  }

1. Which line in the following code would cause a compile error:

public class Hero {

    public static final String LEAGUE = "HERO"; // Option A

    public String powerLookup(int which) {

        String rtn = LEAGUE + ": Flight"; // Option B

        if (which < 0) {

            rtn = LEAGUE + ": Laser Eyes"; // Option C

        }

        return rtn;

    }

    public static void main(String[] args) {

        System.out.println(Hero.LEAGUE); // Option D

         Hero.LEAGUE = "Villain"; // Option F

        Hero ajax = new Hero();

        System.out.println(ajax.powerLookup(-1));

  }

}