REVIEW FOR TEST:

1. Consider the following instance variable and method.

```
private List<String> animals;

public void manipulate()
{
  for (int k = animals.size() - 1; k > 0; k--)
  {
    if (animals.get(k).substring(0, 1).equals("b"))
    {
      animals.add(animals.size() - k, animals.remove(k));
    }
  }
}
```

Assume that `animals` has been instantiated and initialized with the following contents.

```
["bear", "zebra", "bass", "cat", "koala", "baboon"]
```

What will the contents of `animals` be as a result of calling `manipulate` ?

(A) `["baboon", "zebra", "bass", "cat", "bear", "koala"]`
(B) `["bear", "zebra", "bass", "cat", "koala", "baboon"]`
(C) `["baboon", "bear", "zebra", "bass", "cat", "koala"]`
(D) `["bear", "baboon", "zebra", "bass", "cat", "koala"]`
(E) `["zebra", "cat", "koala", "baboon", "bass", "bear"]`

2. Consider the following code segment.

```
int[] oldArray = {1, 2, 3, 4, 5, 6, 7, 8, 9};
int[][] newArray = new int[3][3];

int row = 0;
int col = 0;
for (int value : oldArray)
{
  newArray[row][col] = value;
  row++;
  if ((row % 3) == 0)
  {
    col++;
    row = 0;
  }
}

System.out.println(newArray[0][2]);
```

What is printed as a result of executing the code segment?

(A) 3
(B) 4
(C) 5
(D) 7
(E) 8

3. A rectangular box fits inside another rectangular box if and only if the height, width, and depth of the smaller box are each less than the corresponding values of the larger box. Consider the following three interface declarations that are intended to represent information necessary for rectangular boxes.

I.
```
public interface RBox
{
  /** @return the height of this RBox */
  double getHeight();

  /** @return the width of this RBox */
  double getWidth();

  /** @return the depth of this RBox */
  double getDepth();
}
```

II.
```
public interface RBox
{
  /** @return true if the height of this RBox is less than the height of other;
   *          false otherwise
   */
  boolean smallerHeight(RBox other);

  /** @return true if the width of this RBox is less than the width of other;
   *          false otherwise
   */
  boolean smallerWidth(RBox other);

  /** @return true if the depth of this RBox is less than the depth of other;
   *          false otherwise
   */
  boolean smallerDepth(RBox other);
}
```

III.
```
public interface RBox
{
  /** @return the surface area of this RBox */
  double getSurfaceArea();

  /** @return the volume of this RBox */
  double getVolume();
}
```

Which of the interfaces, if correctly implemented by a `Box` class, would be sufficient functionality for a user of the `Box` class to determine if one `Box` can fit inside another?

(A) I only

(B) II only

(C) III only

(D) I and II only

(E) I, II, and III

4. Refer to the following declarations

```
String[] colors = {"red", "green", "black"};
List<String> colorList = new ArrayList<String>();
```

Which of the following correctly assigns the elements of the colors array to colorList? The order in colorList should be reverse of array colors.

A.
```
int i=0;
for (String s : colorList) {
 s = colors[i];
 i++;
}
```

B.
```
for (int i = colors.length-1; i >=0; i--)
    colorList.add(colors[i]);
```

C.
```
for (String s: colors)
    colorList.set(0,s);
```

D.
```
for (String s: colors)
    colorList.add(s);
```

5. Consider the following data field and method.

```
private String[][] puzzle;

public void adjust()
{
  for(int row = 0; row < puzzle.length; row++)
  {
    String temp = puzzle[row][puzzle[row].length-1];
    for(int col = puzzle[row].length - 1; col > row; col--)
      puzzle[row][col] = puzzle[row][col-1];
    puzzle[row][row] = temp;
  }
}
```

Assume that puzzle contains the following single letter strings. Note that puzzle[0][5] = "F".

```
A  B  C  D  E  F
G  H  I  J  K  L
M  N  O  P  Q  R
S  T  U  V  W  X
```

What values does puzzle contain after the call adjust()?

A.
```
A  B  C  D  E  F
G  H  I  J  K  L
M  N  O  P  Q  R
S  T  U  V  W  X
```

B.
```
B  C  D  E  F  A
H  I  J  K  L  G
N  O  P  Q  R  M
T  U  V  W  X  S
```

C.
```
F  A  B  C  D  E
L  G  H  I  J  K
R  M  N  O  P  Q
X  S  T  U  V  W
```

D.
```
F  A  B  C  D  E
G  L  H  I  J  K
M  N  R  O  P  Q
S  T  U  X  V  W
```

E.
```
A  B  C  D  E  F
G  H  I  J  L  K
M  N  O  R  P  Q
S  T  X  U  V  W
```

6. Consider the following instance variable and method.

```
private String[] nameList;

public int getIt(String name){
  int val = 0;

  for(String fromList: nameList){
    if(fromList.equals(name))
      val++;
  }
  return val;
}
```

Assume that `nameList` has been appropriately initialized. Which of the following best describes what, if anything, is returned by call `getIt("Dennis")`?

A. The index of the first occurrence of `"Dennis"` in `nameList`.
B. The index of the last occurrence of `"Dennis"` in `nameList`.
C. The number of times `"Dennis"` occurs in `nameList`.
D. The number of items not equal to `"Dennis"` in `nameList`.
E. Nothing because the code is incorrect and will not compile.

7. Consider the following declarations.

```
public class ColorBox {
  public ColorBox(){
    System.out.print("black ");
  }

  public void showColor() {
    System.out.print("red ");
  }
}

public class BlueGreenBox extends ColorBox {
  public BlueGreenBox() {
    System.out.print("blue ");
    super();
  }

  public void showColor() {
    System.out.print("green ");
    super.showColor();
  }
}
```

The following statements occur in a client method:

```
ColorBox box = new BlueGreenBox();
box.showColor();
```

What is happens when these classes and the client are compiled and executed?

A. `blue black green red` is printed
B. `black blue green red` is printed
C. `black blue black green red` is printed
D. `black blue black red green red` is printed
E. The classes and code fail to compile correctly

8. Assume that the following partial declarations have been made, with default constructors for the classes.

```
public interface Computer

public class Widget implements Computer

public class Thingy extends Widget
```

Consider the following declarations.

```
I.    Widget myThing = new Thingy();

II.   Controller myWidge = new Thingy();

III.  Controller myControl = new Controller();
```

Which of these declarations will compile correctly?

A. I only
B. II only
C. III only
D. I and II
E. II and III

9. Consider the following description
An estate has a house and has a garden. A garden has flowers. A crocus is a flower and a daffodil is a flower.
Which of the following partial declarations would be the best choice for representing the relationships among these things?

```
A. public class House
   public class Garden
   {
     private ArrayList<Flower> myFlowers;
       . . .
   }

   public class Estate
   {
      private House myHouse;
      private Garden myGarden;
       . . .
   }
   public class Flower
   public class Crocus extends Flower
   public class Daffodil extends Flower

B. public class House
   public class Garden
   {
     private ArrayList<Flower> myFlowers;
       . . .
   }
   public class Estate extends House
   {
      private Garden myGarden;
       . . .
   }
   public class Flower
   public class Crocus extends Flower
```

```
   public class Daffodil extends Flower

C. public class House
   public class Garden
   {
     private ArrayList<Flower> myFlowers;
       . . .
   }

   public class Estate
   {
      private House myHouse;
      private Garden myGarden;
       . . .
   }

   public class Flower
   {
      private boolean isCrocus;
      private boolean isDaffodil;
   }

D. public class House
   public class Garden
   {
     private ArrayList<Flower> myFlowers;
       . . .
   }

   public class Estate extends House
   {
      private Garden myGarden;
       . . .

   }

   public class Flower
   {
      private boolean isCrocus;
      private boolean isDaffodil;
   }

E. public class House
   public class Garden

   public class Estate
   {
      private House myHouse;
      private Garden myGarden;
       . . .
   }
   public class Flower
   {
      private Garden myGarden;
   }

   public class Crocus extends Flower
   public class Daffodil extends Flower
```

10. Consider the following method.

```
public ArrayList<Integer> mystery(int n)
{
  ArrayList<Integer> seq = new ArrayList<Integer>();

  for (int k = 0; k <= n; k++)
    seq.add(new Integer(k * k + 3));

  return seq;
}
```

Which of the following is printed as a result of executing the following statement?

```
System.out.println(mystery(6));
```

(A) `[3, 4, 7, 12, 19, 28]`

(B) `[3, 4, 7, 12, 19, 28, 39]`

(C) `[4, 7, 12, 19, 28, 39]`

(D) `[39, 28, 19, 12, 7, 4]`

(E) `[39, 28, 19, 12, 7, 4, 3]`

11. Consider the following code segment.

```
int[] arr = {7, 2, 5, 3, 0, 10};
for (int k = 0; k < arr.length - 1; k++)
{
  if (arr[k] > arr[k + 1])
    System.out.print(k + "  " + arr[k] + "  ");
}
```

What will be printed as a result of executing the code segment?

(A) `0  2  2  3  3  0`

(B) `0  7  2  5  3  3`

(C) `0  7  2  5  5  10`

(D) `1  7  3  5  4  3`

(E) `7  2  5  3  3  0`

12. Assume that `myList` is an `ArrayList` that has been correctly constructed and populated with objects. Which of the following expressions produces a valid random index for `myList` ?

(A) `(int)( Math.random() * myList.size() ) - 1`

(B) `(int)( Math.random() * myList.size() )`

(C) `(int)( Math.random() * myList.size() ) + 1`

(D) `(int)( Math.random() * (myList.size() + 1) )`

(E) `Math.random(myList.size())`

3. Which of the following represents an acceptable definition of an interface?

select

```
public class Rectangle {
     private double length,width;
     public Rectangle(double l,double w) {
          length = l;
          width = w;
     }
     public double getArea() {
          return length * width;
     };
     public double getPerimeter() {
          return 2 * length + 2 * width;
     }

}
```

select

```
public interface Person {
    private String fName;
    private String lName;
   public String toString();
}
```

select

```
public interface Shape2D {
     public double getArea();
     public double getPerimeter();
}
```

select

```
public interface Person {
    private String fName;
    private String lName;
   public String toString() {
        return fName + " " + lName;
  }
}
```

select

```
public interface Rectangle {
     private double length,width;
     public Rectangle(double l,double w) {
          length = l;
          width = w;
     }
     public double getArea();
     public double getPerimeter();
}
```

1. B 2. D 3. D 4. B 5. D 6. C 7. E 8. D 9. A 10. B 11. B 12. B 13. C