1. Consider the following method.

```
public int someMethod(int val)
{
   for (int i = 2; i < 7; i++)
   {
      if((val + i) % 2 == 0)
      {
         val += 3;
      }
   }
   return val;
}
```

What value is returned by the call `someMethod(13)`?

(A) `17`
(B) `25`
(C) `28`
(D) `31`
(E) Nothing is returned. There is a compile-time error.

2. Consider the following code segment.

```
int num1 = 2;
int num2 = 13;
int result = 4;

if ((num1 < 5) && (num2 < 5))
   result = num1 - num2;
else if ((num1 == 2) && (num2 < 2))
   result = num2 - num1;
else
   result = num1 + num2;
System.out.println(result);
```

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

(A) `-11`
(B) `4`
(C) `11`
(D) `13`
(E) `15`

3. Assume `list` is an `ArrayList<Integer>` that has been correctly constructed and populated with the following items.

```
[13, 7, 0, 5, 12, 6, 10]
```

Consider the following method.

```
public int calculate(ArrayList<Integer> numbers)
{
   int sum = 0;

   for (Integer n : numbers)
   {
      if (n - 8 > 0)
      {
         sum = sum + n;
      }
   }
   return sum;
}
```

What value is returned by the call `calculate(list)`?

(A) `10`
(B) `11`
(C) `13`
(D) `35`
(E) `45`

4. Consider the following class declarations.

```
public class Planet
{
   private String name;
   private double mass;
   private int position;

   public Planet()
   {  /* implementation not shown  */  }

   public Planet(String name)
   {  /* implementation not shown  */  }

   public Planet(String name, int position)
   {  /* implementation not shown  */  }

}

public class DwarfPlanet extends Planet
{
   private double distance;
   public DwarfPlanet(String name)
   {  /* implementation not shown  */  }
}
```

Which of the following declarations compiles without error?

I. `Planet mars = new Planet();`
II. `Planet pluto = new DwarfPlanet("Pluto");`
III. `Planet ceres = new DwarfPlanet();`

(A) I only
(B) II only
(C) I and II only
(D) I and III only
(E) I, II, and III

5. Consider the following code segment.

```
List<String> supernatural = new ArrayList<String>();

supernatural.add("Vampire");
supernatural.add("Werewolf");
supernatural.add("Ghost");
supernatural.set(0, "Zombie");
supernatural.add(2, "Mummy”);
supernatural.add("Witch");
supernatural.remove(3);
System.out.println(supernatural);
```

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

(A) `[Zombie, Werewolf, Mummy, Witch]`
(B) `[Zombie, Werewolf, Ghost, Witch]`
(C) `[Zombie, Werewolf, Mummy, Ghost]`
(D) `[Zombie, Vampire, Werewolf, Mummy, Ghost]`
(E) `[Zombie, Vampire, Werewolf, Mummy, Witch]`

6. Consider the following interface and class declarations.

```
public interface Polygon
{
   /**
    * @return whether the polygon contains the point defined by (x, y)
    */
   boolean contains(int x, int y);
}

public class ClickableZones
{
   private ArrayList<Polygon> zones;

   public int getNumberOfZones(int x, int y)
   {
      int total = 0;
      for (Polygon p : zones)
      {
         /*  missing code  */
      }
      return total;
   }

   /* Additional implementation not shown */
}
```

Which of the following could replace `/* missing code */` so that the method `getNumberOfZones` returns the number of `Polygons` that contain the coordinate point defined by `x` and `y`?

(A)
```
if (polygon contains x, y)
   return true;
```

(B)
```
if (p.contains(x, y))
   return true;
```

(C)
```
if (!p.contains(x, y))
   return false;
```

(D)
```
if (Polygon.contains(x, y))
   total++;
```

(E)
```
if (p.contains(x, y))
   total++;
```

7. Consider the following partial class declaration.

```
public class Park
{
   private String name;
   private boolean playground;
   private int acres;

   public Park(String myName, boolean myPlayground, int myAcres)
   {
      name = myName;
      playground = myPlayground;
      acres = myAcres;
   }

   public String getName()
   {  return name;  }

   public boolean hasPlayground()
   {  return playground;  }

   public int getAcres()
   {  return acres;  }

   /* Additional implementation not shown */
}
```

Assume that the following declaration has been made in the main method of another class.

```
Park park = new Park("Central", true, 300);
```

Which of the following statements compiles without error?

(A) `int num = park.acres;`

(B) `String name = central.getName();`

(C) `boolean play = park.hasPlayground();`

(D) `int num = park.getAcres(acres);`

(E) `park.hasPlayground = true;`

8. Consider the following code segment.

```
String idk = "mnomnomno";
for (int i = 0; i < idk.length(); i++)
{
   if (idk.substring(i, i + 1).equals("m"))
   {
      idk = idk.substring(0, i) + idk.substring(i + 1, idk.length());
   }
}
System.out.println(idk);
```

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

(A) `mmm`
(B) `nonono`
(C) `mnomno`
(D) `nomnono`
(E) `mnomnomno`

9. Consider the following method.

```
public int loopy(int n)
{
   if (n % 7 == 0)
   {
      return n;
   }
   return loopy(n + 3) + 2;
}
```

What value is returned by the call `loopy(12)`?

(A) `12`
(B) `21`
(C) `23`
(D) `27`
(E) `29`

10. Consider the following class declarations.

```
public class Letter
{
   private String letter = "letter";

   public String toString()
   {  return letter;  }

   /* Additional implementation not shown */
}

public class ALetter extends Letter
{
   private String letter = "a";

   public String toString()
   {  return letter;  }

   /* Additional implementation not shown */
}

public class BLetter extends Letter
{
   private String letter = "b";

   public String toString()
   {  return letter;  }

   /* Additional implementation not shown */
}

public class CapALetter extends ALetter
{
   private String letter = "A";

   /* Additional implementation not shown */
}
```

Consider the following code segment.

```
Letter x = new ALetter();
Letter y = new BLetter();
ALetter z = new CapALetter();

System.out.print(x);
System.out.print(y);
System.out.print(z);
```

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

(A) `abA`
(B) `aba`
(C) `letterlettera`
(D) `letterletterletter`
(E) Nothing is printed. There is a compile-time error.

11. Consider the following method.

```
public String lengthen(String word)
{
   int index = 0;
   while (index < word.length())
   {
      word = word + word.substring(index, index + 1);
      index += 2;
   }
   return word;
}
```

What is returned by the call `lengthen("APCS")`?

(A) `"APCS"`
(B) `"APCSACAA"`
(C) `"APCSAPCS"`
(D) Nothing is returned. Run-time error: `StringIndexOutOfBoundsException`
(E) Nothing is returned. The call will result in an infinite loop.

12. Consider the following code segment.

```
int[] array = {-3, 0, 2, 4, 5, 9, 13, 1, 5};
for (int n = 1; n < array.length - 1; n++)
{
   if (array[n] - array[n - 1] <= array[n] - array[n + 1])
      System.out.print(array[n] + "  ");
}
```

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

(A) `1`
(B) `13`
(C) `13 1`
(D) `2 4 5`
(E) `2 4 5 9`

13. Assume that `k`, `m`, and `n` have been declared and correctly initialized with `int` values. Consider the following statement.

```
boolean b1 = (n >= 4) || ((m == 5 || k < 2) && (n > 12));
```

For which statement below does `b2 = !b1` for all values of `k`, `m`, and `n`?

(A) `boolean b2 = (n >= 4) && ((m == 5 && k < 2) || (n > 12));`
(B) `boolean b2 = (n < 4) || ((m != 5 || k >= 2) && (n <= 12));`
(C) `boolean b2 = (n < 4) && (m != 5) && (k >= 2) || (n <= 12);`
(D) `boolean b2 = (m == 5 || k < 2) && (n > 12);`
(E) `boolean b2 = (n < 4);`

14. Consider the following code segment.

```
int[] ray = new int[11];

for (int i = 0; i < ray.length; i++)
{
   ray[i] = i * 2;
}

for (int m = 0; m < 5; m++)
{
   for (int n = 0; n < 7; n += 2)
   {
      if (m + n > 8)
      {
         System.out.print(ray[m + n]);
      }
   }
}
```

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

(A) Nothing is printed. Runtime error: `ArrayIndexOutOfBounds`
(B) `18`
(C) `181620`
(D) `68`
(E) `1820`

15. Consider the following method.

```
public int mystery(String code, int index)
{
   if (code.indexOf("c") == index)
   {
      return index;
   }
   return mystery(code.substring(2), index + 1);
}
```

Assume that the string `codeword` has been declared and initialized as follows.

```
String codeword = "advanced placement";
```

What value is returned by the call `mystery(codeword, 9)`?

(A) `5`
(B) `6`
(C) `7`
(D) Nothing is returned. Infinite recursion causes a stack overflow error.
(E) Nothing is returned. Run-time error: `StringIndexOutOfBoundsException`

16. Consider the following method.

```
public void switcheroo(int num, int index, int[] nums)
{
   int temp = num;
   num = nums[index];
   nums[index] = temp;
   index++;
}
```

Consider the following code segment.

```
int[] val = {5, 7, 4, -2, 8, 12};
int num = 10;
int index = 3;
switcheroo(num, index, val);

System.out.println("num = " + num + "  val[" + index + "] = " + val[index]);
```

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

(A) `num = 10  val[3] = 10`
(B) `num = 10  val[3] = -2`
(C) `num = 10  val[4] = 8`
(D) `num = -2  val[3] = 10`
(E) `num = -2  val[4] = 8`

17. Consider the following code segment.

```
for (int h = 2; h <= 6; h += 2)
{
   for (int k = 30; k > 0; k -= 10)
   {
      System.out.print(h + k + "    ");
   }
}
```

Consider these additional code segments.

I.
```
int num = 32;
int count = 0;
for (int i = 0; i < 9; i++)
{
   System.out.print(num + "    ");
   num += 2;
   if (count % 3 == 0)
   {
      count = 0;
      num -= 14;
   }
}
```

II.
```
int num = 32;
while (num < 38)
{
   System.out.print(num + "    ");
   num -= 10;
   if (num < 10)
   {
      num += 32;
   }
}
```

III.
```
for (int h = 0; h <= 3; h++)
{
   for (int k = 30; k > 0; k -= 10)
   {
      System.out.print(k + h + "    ");
   }
}
```

Which of the code segments produce the same output as the original code segment?

(A) I only
(B) II only
(C) III only
(D) II and III only
(E) I, II, and III

18. Consider the following method.

```
public void mystery (int[] array)
{
   for (int i = 1; i < array.length; i++)
   {
      int j;
      int key = array[i];
      for (j = i - 1; j >= 0 && array[j] > key; j--)
      {
         array[j + 1] = array[j];
      }
      array[j + 1] = key;
   }
}
```

The method above could be best described as an implementation of which of the following?
(A) Insertion Sort
(B) Binary Search
(C) Selection Sort
(D) Merge Sort
(E) Sequential Sort

19. Consider the following statement.

```
int number = (int)(Math.random() * 21 + 13);
```

After executing the statement, what are the possible values for the variable `number`?
(A) All integers from 13 to 21 (inclusive).
(B) All real numbers from 13 to 34 (not including 34).
(C) All integers from 13 to 34 (inclusive).
(D) All integers from 13 to 33 (inclusive).
(E) All real numbers from 0 to 21 (not including 21).

20. Consider the following class declaration.

```
public class City
{
   private String name;
   private int population;

   public City(String myName, int myPop)
   {
      name = myName;
      population = myPop;
   }

   public String getName()
   {  return name;  }

   public int getPopulation()
   {  return population;  }

   /*  Additional implementation not shown */
}
```

Assume `ArrayList<City> cities` has been properly instantiated and populated with `City` objects.

Consider the following code segment.

```
int maxPop = Integer.MIN_VALUE;
for (int i = 0; i < cities.size(); i++)
{
   /* missing code */
}
```

Which of the following should replace `/* missing code */` so that, after execution is complete, `maxPop` will contain the largest population that exists in the `ArrayList`?

(A)
```
City temp = cities[i];
if (temp.getPopulation() > maxPop)
{
   maxPop = temp.getPopulation();
}
```

(B)
```
City temp = cities.get(i);
if (temp.population > maxPop)
{
   maxPop = temp.population;
}
```

(C)
```
if (cities.get(i + 1).getPopulation() > cities.get(i).getPopulation())
{
   maxPop = cities.get(i + 1).getPopulation();
}
```

(D)
```
if (cities.get(i).getPopulation() > maxPop)
{
   maxPop = cities.get(i).getPopulation();
}
```

(E) `maxPop` should have been set to `Integer.MAX_VALUE`. This cannot work as written.

STOP. End of Part I.