Section I

A teacher put three bonus questions on a test and awarded 5 extra points to anyone who answered all three bonus questions correctly and no extra points otherwise. Assume that the `boolean` variables `bonusOne`, `bonusTwo`, and `bonusThree` indicate whether a student has answered the particular question correctly. Each variable was assigned `true` if the answer was correct and `false` if the answer was incorrect.

Which of the following code segments will properly update the variable `grade` based on a student's performance on the bonus questions?

I.
```
if (bonusOne && bonusTwo && bonusThree)
  grade += 5;
```

II.
```
if (bonusOne || bonusTwo || bonusThree)
  grade += 5;
```

III.
```
if (bonusOne)
  grade += 5;
if (bonusTwo)
  grade += 5;
if (bonusThree)
  grade += 5;
```

(A) I only

(B) II only

(C) III only

(D) I and III

(E) II and III

GO ON TO THE NEXT PAGE

2)

Assume that an array of integer values has been declared as follows and has been initialized.

```
int[] arr = new int[10];
```

Which of the following code segments correctly interchanges the value of `arr[0]` and `arr[5]` ?

(A)
```
arr[0] = 5;
arr[5] = 0;
```

(B)
```
arr[0] = arr[5];
arr[5] = arr[0];
```

(C)
```
int k = arr[5];
arr[0] = arr[5];
arr[5] = k;
```

(D)
```
int k = arr[0];
arr[0] = arr[5];
arr[5] = k;
```

(E)
```
int k = arr[5];
arr[5] = arr[0];
arr[0] = arr[5];
```

Section I

4. Consider the following code segment.

```
ArrayList<String> items = new ArrayList<String>();
items.add("A");
items.add("B");
items.add("C");
items.add(0, "D");
items.remove(3);
items.add(0, "E");
System.out.println(items);
```

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

(A) [A, B, C, E]

(B) [A, B, D, E]

(C) [E, D, A, B]

(D) [E, D, A, C]

(E) [E, D, C, B]

---

5. When designing a class hierarchy, which of the following should be true of a superclass?

(A) A superclass should contain the data and functionality that are common to all subclasses that inherit from the superclass.

(B) A superclass should be the largest, most complex class from which all other subclasses are derived.

(C) A superclass should contain the data and functionality that are only required for the most complex class.

(D) A superclass should have public data in order to provide access for the entire class hierarchy.

(E) A superclass should contain the most specific details of the class hierarchy.

4)

8. Consider the following instance variable and incomplete method. The method `calcTotal` is intended to return the sum of all values in `vals`.

```
private int[] vals;

public int calcTotal()
{
  int total = 0;

  /* missing code */

  return total;
}
```

Which of the code segments shown below can be used to replace `/* missing code */` so that `calcTotal` will work as intended?

I.
```
for (int pos = 0; pos < vals.length; pos++)
{
  total += vals[pos];
}
```

II.
```
for (int pos = vals.length; pos > 0; pos--)
{
  total += vals[pos];
}
```

III.
```
int pos = 0;
while (pos < vals.length)
{
  total += vals[pos];
  pos++;
}
```

(A) I only

(B) II only

(C) III only

(D) I and III

(E) II and III

GO ON TO THE NEXT PAGE.

5)

10. Consider the following method.

```
public void numberCheck(int maxNum)
{
  int typeA = 0;
  int typeB = 0;
  int typeC = 0;

  for (int k = 1; k <= maxNum; k++)
  {
    if (k % 2 == 0 && k % 5 == 0)
      typeA++;
    if (k % 2 == 0)
      typeB++;
    if (k % 5 == 0)
      typeC++;
  }

  System.out.println(typeA + " " + typeB + " " + typeC);
}
```

What is printed as a result of the call `numberCheck(50)` ?

(A) `5 20 5`

(B) `5 20 10`

(C) `5 25 5`

(D) `5 25 10`

(E) `30 25 10`

6)

11. Consider the following method that is intended to modify its parameter `nameList` by replacing all occurrences of `name` with `newValue`.

```
public void replace(ArrayList<String> nameList,
                    String name, String newValue)
{
  for (int j = 0; j < nameList.size(); j++)
  {
    if ( /* expression */ )
    {
      nameList.set(j, newValue);
    }
  }
}
```

Which of the following can be used to replace `/* expression */` so that `replace` will work as intended?

(A) `nameList.get(j).equals(name)`

(B) `nameList.get(j) == name`

(C) `nameList.remove(j)`

(D) `nameList[j] == name`

(E) `nameList[j].equals(name)`

14. Consider the following instance variables and incomplete method that are part of a class that represents an item. The variables `years` and `months` are used to represent the age of the item, and the value for `months` is always between 0 and 11, inclusive. Method `updateAge` is used to update these variables based on the parameter `extraMonths` that represents the number of months to be added to the age.

```
private int years;
private int months;  // 0 <= months <= 11

// precondition: extraMonths >= 0
public void updateAge(int extraMonths)
{
  /* body of updateAge */
}
```

Which of the following code segments could be used to replace `/* body of updateAge */` so that the method will work as intended?

I.
```
int yrs = extraMonths % 12;
int mos = extraMonths / 12;
years = years + yrs;
months = months + mos;
```

II.
```
int totalMonths = years * 12 + months + extraMonths;
years = totalMonths / 12;
months = totalMonths % 12;
```

III.
```
int totalMonths = months + extraMonths;
years = years + totalMonths / 12;
months = totalMonths % 12;
```

(A) I only

(B) II only

(C) III only

(D) II and III only

(E) I, II, and III

13. Consider the following method.

```
public String inRangeMessage(int value)
{
  if (value < 0 || value > 100)
    return "Not in range";
  else
    return "In range";
}
```

Consider the following code segments that could be used to replace the body of `inRangeMessage`.

I.
```
if (value < 0)
{
  if (value > 100)
    return "Not in range";
  else
    return "In range";
}
else
  return "In range";
```

II.
```
if (value < 0)
  return "Not in range";
else if (value > 100)
  return "Not in range";
else
  return "In range";
```

III.
```
if (value >= 0)
  return "In range";
else if (value <= 100)
  return "In range";
else
  return "Not in range";
```

Which of the replacements will have the same behavior as the original version of `inRangeMessage` ?

(A) I only

(B) II only

(C) III only

(D) I and III

(E) II and III

GO ON TO THE NEXT PAGE.

9)

10. The following incomplete method is intended to sort its array parameter `arr` in increasing order.

```
// postcondition: arr is sorted in increasing order
public static void sortArray(int[] arr)
{
  int j, k;

  for (j = arr.length - 1; j > 0; j--)
  {
    int pos = j;

    for ( /* missing code */ )
    {
      if (arr[k] > arr[pos])
      {
        pos = k;
      }
    }
    swap(arr, j, pos);
  }
}
```

Assume that `swap(arr, j, pos)` exchanges the values of `arr[j]` and `arr[pos]`. Which of the following could be used to replace `/* missing code */` so that executing the code segment sorts the values in array `arr`?

(A) `k = j - 1; k > 0; k--`

(B) `k = j - 1; k >= 0; k--`

(C) `k = 1; k < arr.length; k++`

(D) `k = 1; k > arr.length; k++`

(E) `k = 0; k <= arr.length; k++`

10)

10. Consider the following instance variables and method that appear in a class representing student information.

```
private int assignmentsCompleted;
private double testAverage;

public boolean isPassing()
{ /* implementation not shown */ }
```

A student can pass a programming course if at least one of the following conditions is met.

- The student has a test average that is greater than or equal to 90.
- The student has a test average that is greater than or equal to 75 and has at least 4 completed assignments.

Consider the following proposed implementations of the `isPassing` method.

I.
```
if (testAverage >= 90)
  return true;
if (testAverage >= 75 && assignmentsCompleted >= 4)
  return true;
return false;
```

II.
```
boolean pass = false;
if (testAverage >= 90)
  pass = true;
if (testAverage >= 75 && assignmentsCompleted >= 4)
  pass = true;
return pass;
```

III.
```
return (testAverage >= 90) ||
       (testAverage >= 75 && assignmentsCompleted >= 4);
```

Which of the implementations will correctly implement method `isPassing`?

(A) I only

(B) II only

(C) I and III only

(D) II and III only

(E) I, II, and III

20. Consider the following instance variables and method that appear in a class representing student information.

```
private int assignmentsCompleted;
private double testAverage;

public boolean isPassing()
{ /* implementation not shown */ }
```

A student can pass a programming course if at least one of the following conditions is met.

- The student has a test average that is greater than or equal to 90.
- The student has a test average that is greater than or equal to 75 and has at least 4 completed assignments.

Consider the following proposed implementations of the `isPassing` method.

I.
```
if (testAverage >= 90)
  return true;
if (testAverage >= 75 && assignmentsCompleted >= 4)
  return true;
return false;
```

II.
```
boolean pass = false;
if (testAverage >= 90)
  pass = true;
if (testAverage >= 75 && assignmentsCompleted >= 4)
  pass = true;
return pass;
```

III.
```
return (testAverage >= 90) ||
       (testAverage >= 75 && assignmentsCompleted >= 4);
```

Which of the implementations will correctly implement method `isPassing`?

(A) I only

(B) II only

(C) I and III only

(D) II and III only

(E) I, II, and III

11)

4. Consider the following code segment.


```
int k = 0;
while (k < 10)
{
  System.out.print((k % 3) + "  ");
  if ((k % 3) == 0)
    k = k + 2;
  else
    k++;
}
```

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

(A) 0 2 1 0 2

(B) 0 2 0 2 0 2

(C) 0 2 1 0 2 1 0

(D) 0 2 0 2 0 2 0

(E) 0 1 2 1 2 1 2

(2)

4. Consider the following method. Method `allEven` is intended to return `true` if all elements in array `arr` are even numbers; otherwise, it should return `false`.

```
public boolean allEven(int[] arr)
{
  boolean isEven = /* expression */ ;

  for (int k = 0; k < arr.length; k++)
  {
    /* loop body */
  }

  return isEven;
}
```

Which of the following replacements for `/* expression */` and `/* loop body */` should be used so that method `allEven` will work as intended?

	`/* expression */`	`/* loop body */`
(A)	`false`	`if ((arr[k] % 2) == 0)` `  isEven = true;`

(B)	`false`	`if ((arr[k] % 2) != 0)`
		`  isEven = false;`
		`else`
		`  isEven = true;`

(C)	`true`	`if ((arr[k] % 2) != 0)` `  isEven = false;`

(D)	`true`	`if ((arr[k] % 2) != 0)`
		`  isEven = false;`
		`else`
		`  isEven = true;`

(E)	`true`	`if ((arr[k] % 2) == 0)`
		`  isEven = false;`
		`else`
		`  isEven = true;`

13)

28. Consider the following code segment.

```
int x = /* some integer value */ ;
int y = /* some integer value */ ;

boolean result = (x < y);

result = ( (x >= y) && !result );
```

Which of the following best describes the conditions under which the value of `result` will be `true` after the code segment is executed?

(A) Only when `x < y`

(B) Only when `x >= y`

(C) Only when `x` and `y` are equal

(D) The value will always be `true`.

(E) The value will never be `true`.

---

29. Consider the following code segment.

```
for (int outer = 0; outer < n; outer++)
{
  for (int inner = 0; inner <= outer; inner++)
  {
    System.out.print(outer + " ");
  }
}
```

If `n` has been declared as an integer with the value 4, what is printed as a result of executing the code segment?

(A) `0 1 2 3`

(B) `0 0 1 0 1 2`

(C) `0 1 2 2 3 3 3`

(D) `0 1 1 2 2 2 3 3 3 3`

(E) `0 0 1 0 1 2 0 1 2 3`

14)

32. Consider the following code segment.

```
int a = 24;
int b = 30;
while (b != 0)
{
  int r = a % b;
  a = b;
  b = r;
}

System.out.println(a);
```

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

(A) 0

(B) 6

(C) 12

(D) 24

(E) 30

GO ON TO THE NEXT PAGE.

15)

29. Consider the following method.

```
public int sol(int lim)
{
  int s = 0;

  for (int outer = 1; outer <= lim; outer++)
  {
    for (int inner = outer; inner <= lim; inner++)
    {
      s++;
    }
  }

  return s;
}
```

What value is returned as a result of the call `sol(10)` ?

(A) 20

(B) 45

(C) 55

(D) 100

(E) 385

16)

34. Consider the following incomplete method. Method `findNext` is intended to return the index of the first occurrence of the value `val` beyond the position `start` in array `arr`.

```
// returns index of first occurrence of val in arr
// after position start;
// returns arr.length if val is not found
public int findNext(int[] arr, int val, int start)
{
  int pos = start + 1;

  while ( /* condition */ )
    pos++;

  return pos;
}
```

For example, consider the following code segment.

```
int[] arr = {11, 22, 100, 33, 100, 11, 44, 100};

System.out.println(findNext(arr, 100, 2));
```

The execution of the code segment should result in the value 4 being printed.

Which of the following expressions could be used to replace `/* condition */` so that `findNext` will work as intended?

(A) `(pos < arr.length) && (arr[pos] != val)`

(B) `(arr[pos] != val) && (pos < arr.length)`

(C) `(pos < arr.length) || (arr[pos] != val)`

(D) `(arr[pos] == val) && (pos < arr.length)`

(E) `(pos < arr.length) || (arr[pos] == val)`

GO ON TO THE NEXT PAGE.

17)

55. Consider the following code segments.

I.
```
int k = 1;
while (k < 20)
{
  if (k % 3 == 1)
    System.out.print( k + "  ");

  k = k + 3;
}
```

II.
```
for (int k = 1; k < 20; k++)
{
  if (k % 3 == 1)
    System.out.print( k + "  ");
}
```

III.
```
for (int k = 1; k < 20; k = k + 3)
{
  System.out.print( k + "  ");
}
```

Which of the code segments above will produce the following output?

```
1  4  7  10  13  16  19
```

(A) I only

(B) II only

(C) I and II only

(D) II and III only

(E) I, II, and III

16

36. Consider the following two methods that appear within a single class.

```
public void changeIt(int[] list, int num)
{
  list = new int[5];
  num = 0;

  for (int x = 0; x < list.length; x++)
    list[x] = 0;
}


public void start()
{
  int[] nums = {1, 2, 3, 4, 5};
  int value = 6;

  changeIt(nums, value);

  for (int k = 0; k < nums.length; k++)
    System.out.print(nums[k] + " ");

  System.out.print(value);
}
```

What is printed as a result of the call `start()` ?

(A) `0 0 0 0 0 0`

(B) `0 0 0 0 0 6`

(C) `1 2 3 4 5 6`

(D) `1 2 3 4 5 0`

(E) `changeIt` will throw an exception.

GO ON TO THE NEXT PAGE.

19) Consider the following declaration of the class `NumSequence`, which has a constructor that is intended to initialize the instance variable `seq` to an `ArrayList` of `numberOfValues` random floating-point values in the range [0.0, 1.0).

```
public class NumSequence
{
  private ArrayList<Double> seq;

  // precondition:  numberOfValues > 0
  // postcondition: seq has been initialized to an ArrayList of
  //                length numberOfValues; each element of seq
  //                contains a random Double in the range [0.0, 1.0)
  public NumSequence(int numberOfValues)
  {
    /* missing code */
  }
}
```

Which of the following code segments could be used to replace `/* missing code */` so that the constructor will work as intended?

I.
```
ArrayList<Double> seq = new ArrayList<Double>();
for (int k = 0; k < numberOfValues; k++)
  seq.add(new Double(Math.random()));
```

II.
```
seq = new ArrayList<Double>();
for (int k = 0; k < numberOfValues; k++)
  seq.add(new Double(Math.random()));
```

III.
```
ArrayList<Double> temp = new ArrayList<Double>();
for (int k = 0; k < numberOfValues; k++)
  temp.add(new Double(Math.random()));

seq = temp;
```

(A) II only

(B) III only

(C) I and II

(D) I and III

(E) II and III

AP CS 2009 1/2 MC