unit6_final_review

1. 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?

unit6_final_review

(A)

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

(B)

/* expression */	/* loop body */

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

(C)

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

unit6_final_review

(D)

/* expression */	/* loop body */

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

(E)

/* expression */	/* loop body */

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

2. Consider the following instance variable and incomplete method. The method is intended to return a string from the array `words` that would be last alphabetically.

```
private String[] words;

public String findLastWord()
{
     /* missing implementation */
}
```

Assume that `words` has been initialized with one or more strings containing only lowercase letters. Which of the following code segments can be used to replace `/*` *missing implementation* `*/` so that `findLastWord` will work as intended?

unit6_final_review

(A)
```
int maxIndex = 0;
for (int k = 0; k < words.length; k++)
{
     if (words[k].compareTo(maxIndex) > 0)
     {
         maxIndex = k;
     }
}
return words[maxIndex];
```

(B)
```
int maxIndex = 0;
for (int k = 1; k <= words.length; k++)
{
     if (words[k].compareTo(words[maxIndex]) > 0)
     {
         maxIndex = k;
     }
}
return words[maxIndex];
```

(C)
```
int maxIndex = 0;
for (int k = 1; k < words.length; k++)
{
     if (words[k].compareTo(words[maxIndex]) > 0)
     {
         maxIndex = k;
     }
}
return maxIndex;
```

(D)
```
String maxWord = words[0];
for (int k = 1; k < words.length; k++)
{
     if (words[k].compareTo(maxWord) > 0)
     {
         maxWord = k;
     }
}
return maxWord;
```

(E)
```
String maxWord = words[0];
for (int k = 1; k < words.length; k++)
{
     if (words[k].compareTo(maxWord) > 0)
     {
         maxWord = words[k];
     }
}
return maxWord;
```

3. In the following code segment, assume that the string `str` has been properly declared and initialized. The code segment is intended to print the number of strings in the array `animals` that have `str` as a substring.

```
String[] animals = {"horse", "cow", "goat", "dog", "cat", "mouse"};
int count = 0;
for (int i = 0; i <= animals.length; i++)
{
     if (animals[i].indexOf(str) >= 0)
     {
         count++;
     }
}
System.out.println(count);
```

The code segment does not work as intended. Which of the following changes should be made so the code segment works as intended?

(A) The Boolean expression in the `for` loop header should be changed to `i < animals.length`.

(B) The Boolean expression in the `for` loop header should be changed to `i < animals.length - 1`.

(C) The Boolean expression in the `for` loop header should be changed to `i < animals[i].length`.

(D) The condition in the `if` statement should be changed to `animals[i].equals(str)`.

(E) The condition in the `if` statement should be changed to `animals[i].substring(str)`.

4. Consider the following incomplete method that is intended to return an array that contains the contents of its first array parameter followed by the contents of its second array parameter.

```
public static int[] append(int[] a1, int[] a2)
{
  int[] result = new int[a1.length + a2.length];

  for (int j = 0; j < a1.length; j++)
    result[j] = a1[j];

  for (int k = 0; k < a2.length; k++)
    result[ /* index */ ] = a2[k];

  return result;
}
```

Which of the following expressions can be used to replace /* index */ so that append will work as intended?

(A) j

(B) k

(C) k + a1.length - 1

(D) k + a1.length

(E) k + a1.length + 1

unit6_final_review

5. Assume that the array arr has been defined and initialized as follows.

```
int[] arr = /* initial values for the array */ ;
```

Which of the following will correctly print all of the odd integers contained in arr but none of the even integers contained in arr ?

(A)
```
for (int x : arr)
  if (x % 2 != 0)
    System.out.println(x);
```

(B)
```
for (int k = 1; k < arr.length; k++)
  if (arr[k] % 2 != 0)
    System.out.println(arr[k]);
```

(C)
```
for (int x : arr)
  if (x % 2 != 0)
    System.out.println(arr[x]);
```

(D)
```
for (int k = 0; k < arr.length; k++)
  if (arr[k] % 2 != 0)
    System.out.println(k);
```

(E)
```
for (int x : arr)
  if (arr[x] % 2 != 0)
    System.out.println(arr[x]);
```

6. 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];
}
```

unit6_final_review

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

7. Consider the following code segment.

```
int[] arr = {4, 2, 9, 7, 3};
for (int k : arr)
{
k = k + 10;
System.out.print(k + " ");
}
for (int k : arr)
System.out.print(k + " ");
```

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

(A) 0 1 2 3 4 0 1 2 3 4
(B) 4 2 9 7 3 4 2 9 7 3
(C) 10 11 12 13 14 0 1 2 3 4
(D) 14 12 19 17 13 4 2 9 7 3
(E) 14 12 19 17 13 14 12 19 17 13

unit6_final_review

8. 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

9. Consider the following code segment.

```
int[] arr = {1, 2, 3, 4, 5, 6, 7};

for (int k = 3; k < arr.length - 1; k++)
  arr[k] = arr[k + 1];
```

Which of the following represents the contents of arr as a result of executing the code segment?

(A) {1, 2, 3, 4, 5, 6, 7}

(B) {1, 2, 3, 5, 6, 7}

(C) {1, 2, 3, 5, 6, 7, 7}

(D) {1, 2, 3, 5, 6, 7, 8}

(E) {2, 3, 4, 5, 6, 7, 7}

unit6_final_review

10. Consider the following method.

```
public static int mystery(int[] arr)
{
  int x = 0;

  for (int k = 0; k < arr.length; k = k + 2)
    x = x + arr[k];

  return x;
}
```

Assume that the array nums has been declared and initialized as follows.

int [ ] nums = { 3, 6, 1, 0, 1, 4, 2};

What value will be returned as a result of the call mystery(nums) ?

(A) 5
(B) 6
(C) 7
(D) 10
(E) 17

11. Consider the following method, isSorted, which is intended to return true if an array of integers is sorted in nondecreasing order and to return false otherwise.

```
/** @param data an array of integers
 *  @return true if the values in the array appear in sorted (nondecreasing) order
 */
public static boolean isSorted(int[] data)
{
  /* missing code */
}
```

Which of the following can be used to replace /* missing code */ so that isSorted will work as intended?

I.
```
for (int k = 1; k < data.length; k++)
{
  if (data[k - 1] > data[k])
    return false;
}
return true;
```

II.
```
for (int k = 0; k < data.length; k++)
{
  if (data[k] > data[k + 1])
    return false;
}
return true;
```

III.
```
for (int k = 0; k < data.length - 1; k++)
{
  if (data[k] > data[k + 1])
    return false;
  else
    return true;
}
return true;
```

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

unit6_final_review

12. Consider the following method that is intended to return the sum of the elements in the array key.

```
public static int sumArray(int[] key)
{
  int sum = 0;

  for (int i = 1; i <= key.length; i++)
  {
    /* missing code */
  }

  return sum;
}
```

Which of the following statements should be used to replace / * missing code * / so that sumArray will work as intended?

(A) sum = key [i] ;

(B) sum += key [i - 1] ;

(C) sum += key [i] ;

(D) sum += sum + key[i - 1] ;

(E) sum += sum + key [i] ;

unit6_final_review

13. Consider the following method that is intended to return true if an array of integers is arranged in decreasing order and return false otherwise.

```
/** @param nums an array of integers
* @return true if the values in the array appear in decreasing order
* false otherwise
*/
public static boolean isDecreasing(int[] nums)
{
/* missing code */
}
```

Which of the following can be used to replace /* *missing code* */ so that isDecreasing will work as intended?

I.
```
for (int k = 0; k < nums.length; k++)
{
    if (nums[k] <= nums[k + 1])
        return false;
}
return true;
```

II.
```
for (int k = 1; k < nums.length; k++)
{
```

unit6_final_review

```
        if (nums[k - 1] <= nums[k])
          return false;
      }
      return true;
```

III.
```
      for (int k = 0; k < nums.length - 1; k++)
      {
        if (nums[k] <= nums[k + 1])
          return false;
        else
          return true;
      }
      return true;
```

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

unit6_final_review

14. The following incomplete method is intended to return the largest integer in the array numbers.

```
// precondition:  numbers.length > 0

public static int findMax(int[]numbers)
{
        int posOfMax = O;

        for (int index = 1; index < numbers.length; index++)

        {
          if ( /*condition*/ )
          {
                /* statement */

          }
        }
        return numbers[posOfMax];
}
```

Which of the following can be used to replace /* condition */ and /* statement */ so that findMax will work as intended?

unit6_final_review

(A)

/* condition */	/* statement */
numbers[index] > numbers[posOfMax]	posOfMax = numbers[index];

(B)

/* condition */	/* statement */
numbers[index] > numbers[posOfMax]	posOfMax = index;

(C)

/* condition */	/* statement */
numbers[index] > posOfMax	posOfMax = numbers[index];

(D)

/* condition */	/* statement */
numbers[index] < posOfMax	posOfMax = numbers[index];

(E)

/* condition */	/* statement */
numbers[index] < numbers[posOfMax]	posOfMax = index;

unit6_final_review

15. 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)

16. 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];
```

17. Consider the following method.

```
public static int mystery(int value)
{
     int sum = 0;
     int[] arr = {1, 4, 2, 5, 10, 3, 6, 4};

     for (int item : arr)
     {
         if (item > value)
         {
            sum += item;
         }
     }
     return sum;
}
```

What value is returned as a result of the call `mystery(4)` ?

unit6_final_review

(A) 6
(B) 15
(C) 21
(D) 29
(E) 35

18. The question refer to the following data field and method.

```
private int[] arr;

// precondition: arr.length > 0
public void mystery()

{
  int sl = 0;

  int s2 = 0;

  for (int k = 0; k < arr.length; k++)

  {
    int num = arr[k];

    if ((num > 0) && (num % 2 == 0))

      sl += num;

    else if (num < 0)

      s2 += num;

  }

  System.out.println(s1);

  System.out.println(s2);

}
```

Which of the following best describes the value of s1 output by the method mystery ?

(A) The sum of all positive values in arr
(B) The sum of all positive even values in arr
(C) The sum of all positive odd values in arr
(D) The sum of all values greater than 2 in arr
(E) The sum of all values less than 2 in arr

unit6_final_review

19. The question refer to the following data field and method.

```
private int[] arr;

// precondition: arr.length > 0
public void mystery()

{

  int sl = 0;

  int s2 = 0;

  for (int k = 0; k < arr.length; k++)

  {

    int num = arr[k];

    if ((num > 0) && (num % 2 == 0))

      sl += num;

    else if (num < 0)

      s2 += num;

  }

  System.out.println(s1);

  System.out.println(s2);

}
```

Which of the following best describes the value of s2 output by the method mystery ?

(A) The sum of all positive values in arr

(B) The sum of all positive even values in arr

(C) The sum of all negative values in arr

(D) The sum of all negative even values in arr

(E) The sum of all negative odd values in arr

unit6_final_review

20. Consider an integer array, `nums`, which has been declared and initialized with one or more integer values. Which of the following code segments updates `nums` so that each element contains the square of its original value?

I.

```
int k = 0;
while (k < nums.length)
{
     nums[k] = nums[k] * nums[k];
}
```

II.

```
for (int k = 0; k < nums.length; k++)
{
     nums[k] = nums[k] * nums[k];
}
```

III.

```
for (int n : nums)
{
     n = n * n;
}
```

(A) II only

(B) I and II only

(C) I and III only

(D) II and III only

(E) I, II, and III

21. Consider the following incomplete method, which is intended to return the sum of all the elements in its array parameter.

```
/** Precondition: data.length > 0 */
public static int total(int[] data)
{
     int result = 0;

     /* missing code */

     return result;
}
```

The following code segments have been proposed to replace `/* missing code */`.

I.
```
for (int k = 0; k < data.length; k++)
{
   result += k;
}
```

II.
```
for (int d : data)
{
   result += d;
}
```

III.
```
for (int k = 0; k < data.length; k++)
{
   result += data[k];
}
```

Which of the replacements for `/* missing code */` could be used so that `total` will work as intended?

(A) I only

(B) II only

(C) III only

(D) I and II

(E) II and III