U5-10_4

Name

1. Consider the following instance variable, arr, and incomplete method, partialSum. The method is intended to return an integer array sum such that for all k, sum [k] is equal to arr[0] + arr[1] + ... + arr[k]. For instance, if arr contains the values { 1, 4, 1, 3 }, the array sum will contain the values { 1, 5, 6, 9 }.

```
private int[] arr;
public int[] partialSum()
{
   int[] sum = new int[arr.length];
   for (int j = 0; j < sum.length; j++)
   {
      sum[j] = 0;
   }
   /* missing code */
   return sum;
}
```

The following two implementations of / * *missing code* * / are proposed so that partialSum will work as intended.

Implementation 1

```
for (int j = 0; j < arr.length; j++)
{
   sum[j] = sum[j - 1] + arr[j];
}
```

Implementation 2

```
for (int j = 0; j < arr.length; j++)
{
   for (int k = 0; k <= j; k++)
   {
      sum[j] = sum[j] + arr[k];
   }
}
```

Which of the following statements is true?

(A) Both implementations work as intended, but implementation 1 is faster than implementation 2.

(B) Both implementations work as intended, but implementation 2 is faster than implementation 1.

(C) Both implementations work as intended and are equally fast.

(D) Implementation 1 does not work as intended, because it will cause anArrayIndexOutOfBoundsException.

(E) Implementation 2 does not work as intended, because it will cause anArrayIndexOutOfBoundsException.

2. Consider the following interface and class declarations.

```
public interface Student
{   /* implementation not shown */   }

public class Athlete
{   /* implementation not shown */   }

public class TennisPlayer extends Athlete implements Student
{   /* implementation not shown */   }
```

Assume that each class has a zero-parameter constructor. Which of the following is NOT a valid declaration?

(A) Student a = new TennisPlayer();

(B) TennisPlayer b = new TennisPlayer();

(C) Athlete c = new TennisPlayer();

(D) Student d = new Athlete();

(E) Athlete e = new Athlete();

U5-10_4

3. Consider the following interface and class declarations.

```
public interface Vehicle
{
  /** @return  the mileage traveled by this Vehicle
   */
  double getMileage();
}


public class Fleet
{
  private ArrayList<Vehicle> myVehicles;

  /** @return  the mileage traveled by all vehicles in this Fleet
   */
  public double getTotalMileage()
  {
    double sum = 0.0;

    for (Vehicle v : myVehicles)
    {
      sum += /* expression */ ;
    }

    return sum;
  }

  //  There may be instance variables, constructors, and methods that are not shown.
}
```

Which of the following can be used to replace /* expression */ so that getTotalMileage returns the total of the miles traveled for all vehicles in the fleet?

(A) getMileage (v)

(B) myVehicles [v] .getMileage ()

(C) Vehicle.get (v) .getMileage ()

(D) myVehicles.get (v) .getMileage ()

(E) v.getMileage ()

U5-10_4

4. 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] ;

U5-10_4

5. 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;
```

U5-10_4

(A) I only

(B) II only

(C) III only

(D) I and II only

(E) I and III only

---

6. Consider the following method, which is intended to return the element of a 2-dimensional array that is closest in value to a specified number, val.

```
/** @return the element of 2-dimensional array mat whose value is closest to val */
public double findClosest(double[][] mat, double val)
{
  double answer = mat[0][0];
  double minDiff = Math.abs(answer - val);
  for (double[] row : mat)
  {
    for (double num : row)
    {
      if ( /* missing code */ )
      {
        answer = num;
        minDiff = Math.abs(num - val);
      }
    }
  }
  return answer;
}
```

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

U5-10_4

(A) val - row [num] < minDiff

(B) Math.abs (num - minDiff) < minDiff

(C) val - num < 0.0

(D) Math.abs (num - val) < minDiff

(E) Math.abs (row [num] - val) < minDiff

U5-10_4

7. Consider the following method.

```
/** Removes all occurrences of nameToRemove from nameList.
* @param nameList a list of names
* @param nameToRemove a name to be removed from nameList
*/
public void removeName(List<String> nameList, String nameToRemove)
{
/* missing implementation */
}
```

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

1.
```
for (String name : nameList)
{
if (name.equals(nameToRemove))
name.remove();
}
```

2.
```
for (int k = 0; k < nameList.size(); k++)
{
if (nameList.get(k).equals(nameToRemove))
nameList.remove(k);
}
```

3.
```
for (int k = nameList.size() - 1; k >= 0; k--)
{
if (nameList.get(k).equals(nameToRemove))
nameList.remove(k);
}
```

U5-10_4

(A) I only

(B) II only

(C) III only

(D) II and III only

(E) I, II, and III

U5-10_4

8. Consider the following method.

```
// Precondition: b > 0
public int surprise(int b)
{
if ((b % 2) == 0)
{
if (b < 10)
return b;
else
return ((b % 10) + surprise(b / 10));
}
else
{
if (b < 10)
return 0;
else
return surprise(b / 10);
}
}
```

Which of the following expressions will evaluate to true ?

1.
surprise(146781) == 0

2.
surprise(7754) == 4

3.
surprise(58216) == 16

(A) I only

(B) II only

(C) III only

(D) II and III only

(E) I, II, and III

U5-10_4

9. Consider the following method.

```
/** Precondition: arr.length > 0 */
public static int mystery(int[] arr)
{
  int index = 0;
  int count = 0;
  int m = -1;

  for (int outer = 0; outer < arr.length; outer++)
  {
    count = 0;
    for (int inner = outer + 1; inner < arr.length; inner++)
    {
      if (arr[outer] == arr[inner])
      {
        count++;
      }
    }

    if (count > m)
    {
      index = outer;
      m = count;
    }
  }

  return index;
}
```

Assume that nums has been declared and initialized as an array of integer values. Which of the following best describes the value returned by the call mystery(nums) ?

(A) The maximum value that occurs in nums

(B) An index of the maximum value that occurs in nums

(C) The number of times that the maximum value occurs in nums

(D) A value that occurs most often in nums

(E) An index of a value that occurs most often in nums

U5-10_4

10. Consider the following method.

```
/** Precondition: arr contains only positive values.
 */
public static void doSome(int[] arr, int lim)
{
  int v = 0;
  int k = 0;
  while (k < arr.length && arr[k] < lim)
  {
    if (arr[k] > v)
    {
      v = arr[k]; /* Statement S */
    }
    k++; /* Statement T */
  }
}
```

Assume that doSome is called and executes without error. Which of the following are possible combinations for the value of lim, the number of times *Statement* S is executed, and the number of times *Statement* T is executed?

	Value of lim	Executions of *Statement S*	Executions of *Statement T*
I.	5	0	5
II.	7	4	9
III.	3	5	2

U5-10_4

(A) I only

(B) II only

(C) III only

(D) I and III only

(E) II and III only

---

11. 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) ?

U5-10_4

(A) 5

(B) 6

(C) 7

(D) 10

(E) 17

---

12. Consider the following method.

```
public static void mystery(List<Integer> nums)
{
   for (int k = 0; k < nums.size(); k++)
   {
      if (nums.get(k).intValue() == 0)
      {
         nums.remove(k);
      }
   }
}
```

Assume that a List<Integer> values initially contains the following Integer values.

```
[0, 0, 4, 2, 5, 0, 3, 0]
```

What will values contain as a result of executing mystery(values) ?

U5-10_4

(A) [0, 0, 4, 2, 5, 0, 3, 0]

(B) [4, 2, 5, 3]

(C) [0, 0, 0, 0, 4, 2, 5, 3]

(D) [0, 4, 2, 5, 3]

(E) The code throws an ArrayIndexOutOfBoundsException exception.

---

13. Consider the following method.

```
public int addFun(int n)
{
  if (n <= 0)
return 0;
  if (n == 1)
return 2;
  return addFun(n - 1) + addFun(n - 2);
}
```

What value is returned as a result of the call addFun(6) ?

U5-10_4

(A) 10

(B) 12

(C) 16

(D) 26

(E) 32

---

14. Consider the following method.

```
public String recScramble(String str, int[] positions, int k)
{
if (str == null || str.length() == 0)
return "";

if (str.length() == 1)
return str;
int pos = positions[k];
String nStr = str.substring(pos, pos + 1);
str = str.substring(0, pos) + str.substring(pos + 1);
return nStr + recScramble(str, positions, k + 1);
}
```

Consider the following code segment.

```
int[] indexes = {2, 1, 1};
System.out.println(recScramble("epic", indexes, 0));
```

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

(A) cepi

(B) epci

(C) iecp

(D) iepc

(E) ipce

---

15. Consider the following method.

```
public static int[] operation(int[][] matrix, int r, int c)
{
  int[] result = new int[matrix.length];

  for (int j = 0 ; j < matrix.length ; j++)
  {
    result[j] = matrix[r][j] * matrix[j][c];
  }
  return result;
}
```

The following code segment appears in another method in the same class.

```
int[][] mat = {{3, 2, 1, 4},
               {1, 2, 3, 4},
               {2, 2, 1, 2},
               {1, 1, 1, 1}};

int[] arr = operation(mat, 1, 2);
```

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

(A) {6, 4, 2, 4}

(B) {1, 6, 3, 4}

(C) {4, 3, 6, 1}

(D) {4, 4, 2, 2}

(E) {2, 2, 4, 4}

---

16. Consider the following method.

```
public void mystery(int[] data)
{
  for (int k = 0; k < data.length - 1; k++)
    data[k + 1] = data[k] + data[k + 1];
}
```

The following code segment appears in another method in the same class.

```
int[] values = {5, 2, 1, 3, 8};
mystery(values);
for (int v : values)
  System.out.print(v + " ");
System.out.println();
```

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

(A) 5 2 1 3 8

(B) 5 7 3 4 11

(C) 5 7 8 11 19

(D) 7 3 4 11 8

(E) Nothing is printed because an ArrayIndexOutOfBoundsException is thrown during the execution of method mystery.

---

17. Consider the following method.

```
public static void showMe(int arg)
{
  if (arg < 10)
  {
    showMe(arg + 1);
  }
  else
  {
    System.out.print(arg + " ");
  }
}
```

What will be printed as a result of the call showMe(0) ?

(A) 10

(B) 11

(C) 0 1 2 3 4 5 6 7 8 9

(D) 9 8 7 6 5 4 3 2 1 0

(E) 0 1 2 3 4 5 6 7 8 9 10

U5-10_4

18. Consider the following method.

```
/** Precondition: values has at least one row */
public static int calculate(int[][] values)
{
  int found = values[0][0];
  int result = 0;
  for (int[] row : values)
  {
    for (int y = 0; y < row.length; y++)
    {
      if (row[y] > found)
      {
        found = row[y];
        result = y;
      }
    }
  }
  return result;
}
```

Which of the following best describes what is returned by the calculate method?

(A) The largest value in the two-dimensional array

(B) The smallest value in the two-dimensional array

(C) The row index of an element with the largest value in the two-dimensional array

(D) The row index of an element with the smallest value in the two-dimensional array

(E) The column index of an element with the largest value in the two-dimensional array

U5-10_4

19. Consider the following method.

```
/** Precondition: 0 < numVals <= nums.length */
public static int mystery(int[] nums, int v, int numVals)
{
  int k = 0;

  if (v == nums[numVals - 1])
  {
    k = 1;
  }

  if (numVals == 1)
  {
    return k;
  }
  else
  {
    return k + mystery(nums, v, numVals - 1);
  }
}
```

Which of the following best describes what the call mystery(numbers, val, numbers.length) does? You may assume that variables numbers and val have been declared and initialized.

(A) Returns 1 if the last element in numbers is equal to val; otherwise, returns 0

(B) Returns the index of the last element in numbers that is equal to val

(C) Returns the number of elements in numbers that are equal to val

(D) Returns the number of elements in numbers that are not equal to val

(E) Returns the maximum number of adjacent elements that are not equal to val

U5-10_4

20. Consider the following method.

```
/** @param x an int value such that x >= 0
 */
public void mystery(int x)
{
  System.out.print(x % 10);
  if ((x / 10) != 0)
  {
    mystery(x / 10);
  }
  System.out.print(x % 10);
}
```

Which of the following is printed as a result of the call mystery (1234)?

(A) 1234

(B) 4321

(C) 12344321

(D) 43211234

(E) Many digits are printed due to infinite recursion.