

**U5-10\_3**

Name \_\_\_\_\_

The following questions refer to the following classes:

```
public class First
```

```
{
```

```
    public String name()
```

```
    {
```

```
        return "First";
```

```
    }
```

```
}
```

```
public class Second extends First
```

```
{
```

```
    public void whoRules()
```

```
    {
```

```
        System.out.print(super.name() + " rules");
```

```
        System.out.println(" but " + name() + " is even better");
```

```
    }
```

```
    public String name()
```

```
    {
```

```
        return "Second";
```

```
    }
```

```
}
```



U5-10\_3

---

```
public class Third extends Second  
{  
    public String name()  
    {  
        return "Third";  
    }  
}
```

---

1. Consider the following code segment.

```
Second varSecond = new Second();
```

```
Third varThird = new Third();
```

```
varSecond.whoRules();
```

```
varThird.whoRules();
```

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

- (A) First rules but Second is even better  
First rules but Second is even better
  - (B) First rules but Second is even better  
First rules but Third is even better
  - (C) First rules but Second is even better  
Second rules but Second is even better
  - (D) First rules but Second is even better  
Second rules but Third is even better
  - (E) Second rules but Second is even better  
Second rules but Second is even better
- 



**U5-10\_3**

---

2. Consider the following code segment.

```
String[][] board = new String[5][5];

for (int row = 0; row < 5; row++)
{
    for (int col = 0; col < 5; col++)
    {
        board[row][col] = "O";
    }
}

for (int val = 0; val < 5; val++)
{
    if (val % 2 == 1)
    {
        int row = val;
        int col = 0;
        while (col < 5 && row >= 0)
        {
            board[row][col] = "X";
            col++;
            row--;
        }
    }
}
```

Which of the following represents board after this code segment is executed?



## U5-10\_3

(A)

	0	1	2	3	4
0	X	O	X	O	X
1	O	X	O	X	O
2	X	O	X	O	X
3	O	X	O	X	O
4	X	O	X	O	X

(B)

	0	1	2	3	4
0	O	X	O	X	O
1	X	O	X	O	X
2	O	X	O	X	O
3	X	O	X	O	X
4	O	X	O	X	O

(C)

	0	1	2	3	4
0	X	O	O	O	X
1	O	X	O	X	O
2	O	O	X	O	O
3	O	X	O	X	O
4	X	O	O	O	X

(D)

	0	1	2	3	4
0	O	X	O	O	O
1	O	O	X	O	O
2	X	O	O	X	O
3	O	X	O	O	X
4	O	O	X	O	O

(E)

	0	1	2	3	4
0	O	X	O	X	O
1	X	O	X	O	O
2	O	X	O	O	O
3	X	O	O	O	O
4	O	O	O	O	O



U5-10\_3

---

3. Consider the following code segment.

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

students.add("Alex");
students.add("Bob");
students.add("Carl");

for (int k = 0; k < students.size(); k++)
{
    System.out.print(students.set(k, "Alex") + " ");
}

System.out.println();

for (String str : students)
{
    System.out.print(str + " ");
}
```

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

- ☐ A Alex Alex Alex  
Alex Alex Alex
- ☐ B Alex Alex Alex  
Alex Bob Carl
- ☐ C Alex Bob Carl  
Alex Alex Alex
- ☐ D Alex Bob Carl  
Alex Bob Carl
- ☐ E Nothing is printed because the first print statement will cause a runtime exception to be thrown.
- 



**U5-10\_3**

---

4. Consider the following code segment.

```
List<String> animals = new ArrayList<String>();  
  
animals.add("dog");  
animals.add("cat");  
animals.add("snake");  
animals.set(2, "lizard");  
animals.add(1, "fish");  
animals.remove(3);  
System.out.println(animals);
```

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

- (A) [dog, fish, cat]
- (B) [dog, fish, lizard]
- (C) [dog, lizard, fish]
- (D) [fish, dog, cat]
- (E) The code throws an `ArrayIndexOutOfBoundsException` exception.
- 



**U5-10\_3**

---

5. Consider the following code segment.

```
int[][] mat = new int[3][4];
for (int row = 0; row < mat.length; row++)
{
    for (int col = 0; col < mat[0].length; col++)
    {
        if (row < col)
        {
            mat[row][col] = 1;
        }
        else if (row == col)
        {
            mat[row][col] = 2;
        }
        else
        {
            mat[row][col] = 3;
        }
    }
}
```

What are the contents of mat after the code segment has been executed?



U5-10\_3

---

- (A)  $\{\{2, 1, 1\},$   
     $\{3, 2, 1\},$   
     $\{3, 3, 2\},$   
     $\{3, 3, 3\}\}$
- (B)  $\{\{2, 3, 3\},$   
     $\{1, 2, 3\},$   
     $\{1, 1, 2\},$   
     $\{1, 1, 1\}\}$
- (C)  $\{\{2, 3, 3, 3\},$   
     $\{1, 2, 3, 3\},$   
     $\{1, 1, 2, 3\}\}$
- (D)  $\{\{2, 1, 1, 1\},$   
     $\{3, 2, 1, 1\},$   
     $\{3, 3, 2, 1\}\}$
- (E)  $\{\{1, 1, 1, 1\},$   
     $\{2, 2, 2, 2\},$   
     $\{3, 3, 3, 3\}\}$
- 

6. 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?





**U5-10\_3**

---

- (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
- 

7. 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}
- 



**U5-10\_3**

---

8. 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
- 



**U5-10\_3**

---

9. Consider the following declaration for a class that will be used to represent points in the  $xy$ -coordinate plane.

```
public class Point
{
    private int x;        // x-coordinate of the point
    private int y;        // y-coordinate of the point

    public Point()
    {
        x = 0;
        y = 0;
    }

    public Point(int a, int b)
    {
        x = a;
        y = b;
    }

    // Other methods not shown
}
```

The following incomplete class declaration is intended to extend the above class so that points can be named.

```
public class NamedPoint extends Point
{
    private String name; // name of point

    // Constructors go here

    // Other methods not shown
}
```

Consider the following proposed constructors for this class.

- I. 

```
public NamedPoint()
{
    name = "";
}
```
- II. 

```
public NamedPoint(int d1, int d2, String pointName)
{
    x = d1;
    y = d2;
    name = pointName;
}
```
- III. 

```
public NamedPoint(int d1, int d2, String pointName)
{
    super(d1, d2);
    name = pointName;
}
```

Which of these constructors would be legal for the `NamedPoint` class?



U5-10\_3

---

- ☐ A I only
- ☐ B II only
- ☐ C III only
- ☐ D I and III only
- ☐ E II and III only
- 

Directions: Select the choice that best fits each statement. The following question(s) refer to the following incomplete class declaration.

```
public class TimeRecord
{
    private int hours;
    private int minutes; // 0 ≤ minutes < 60
    /** Constructs a TimeRecord object.
     * @param h the number of hours
     *      Precondition:  $h \geq 0$ 
     * @param m the number of minutes
     *      Precondition:  $0 \leq m < 60$ 
     */
    public TimeRecord(int h, int m)
    {
        hours = h;
        minutes = m;
    }

    /** @return the number of hours
     */
    public int getHours()
    { /* implementation not shown */ }

    /** @return the number of minutes
     *      Postcondition:  $0 \leq minutes < 60$ 
     */
    public int getMinutes()
    { /* implementation not shown */ }

    /** Adds h hours and m minutes to this TimeRecord.
     * @param h the number of hours
     *      Precondition:  $h \geq 0$ 
     * @param m the number of minutes
     *      Precondition:  $m \geq 0$ 
     */
    public void advance(int h, int m)
    {
        hours = hours + h;
        minutes = minutes + m;
        /* missing code */
    }
    // Other methods not shown
}
```



**U5-10\_3**

---

10. Consider the following declaration that appears in a class other than `TimeRecord`.  
`TimeRecord [ ] timeCards = new TimeRecord [100] ;`

Assume that `timeCards` has been initialized with `TimeRecord` objects. Consider the following code segment that is intended to compute the total of all the times stored in `timeCards`.

```
TimeRecord total = new TimeRecord(0,0);  
for (int k = 0; k < timeCards.length; k++)  
{  
    /* missing expression */ ;  
}
```

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

- (A) `timeCards [ k ].advance ( )`
- (B) `total += timeCards [ k ].advance ( )`
- (C) `total.advance(timeCards[k].hours,  
 timeCards[k].minutes)`
- (D) `total.advance(timeCards[k].getHours(),  
 timeCards[k].getMinutes())`
- (E) `timeCards[k].advance(timeCards[k].getHours(),  
 timeCards[k].getMinutes())`
- 

11. Consider the following definition.

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

Which of the following code segments produces the output 123456 ?



U5-10\_3

---

- (A) 

```
for (int[] row : numbers)
{
    for (int n : row)
    {
        System.out.print(n);
    }
}
```
- (B) 

```
for (int[] row : numbers)
{
    for (int n : row)
    {
        System.out.print(row[n]);
    }
}
```
- (C) 

```
for (int rc = 0; rc < numbers.length; rc++)
{
    System.out.print(numbers[rc]);
}
```
- (D) 

```
for (int r = 0; r < numbers[0].length; r++)
{
    for (int c = 0; c < numbers.length; c++)
    {
        System.out.print(numbers[r][c]);
    }
}
```
- (E) 

```
for (int c = 0; c < numbers[0].length; c++)
{
    for (int r = 0; r < numbers.length; r++)
    {
        System.out.print(numbers[r][c]);
    }
}
```

---



## U5-10\_3

12. Consider the following incomplete method that is intended to return a string formed by concatenating elements from the parameter words. The elements to be concatenated start with startIndex and continue through the last element of words and should appear in reverse order in the resulting string.

```
/** Precondition: words.length > 0;
 *              startIndex >= 0
 */
public static String concatWords(String[] words, int startIndex)
{
    String result = "";

    /* missing code */

    return result;
}
```

For example, the following code segment uses a call to the concatWords method.

```
String[] things = {"Bear", "Apple", "Gorilla", "House", "Car"};
System.out.println(concatWords(things, 2));
```

When the code segment is executed, the string "CarHouseGorilla" is printed.

The following three code segments have been proposed as replacements for */\* missing code \*/*.

- I.     for (int k = startIndex; k < words.length; k++)  
        {  
            result += words[k] + words[words.length - k - 1];  
        }
- II.    int k = words.length - 1;  
        while (k >= startIndex)  
        {  
            result += words[k];  
            k--;  
        }
- III.   String[] temp = new String[words.length];  
        for (int k = 0; k <= words.length / 2; k++)  
        {  
            temp[k] = words[words.length - k - 1];  
            temp[words.length - k - 1] = words[k];  
        }
- for (int k = 0; k < temp.length - startIndex; k++)  
        {  
            result += temp[k];  
        }

Which of these code segments can be used to replace */\* missing code \*/* so that concatWords will work as intended?



U5-10\_3

---

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

13. 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?





U5-10\_3

---

- (A) j
- (B) k
- (C) k + a1.length - 1
- (D) k + a1.length
- (E) k + a1.length + 1
- 

14. Consider the following instance variable and method.

```
private int[] arr;

/** Precondition: arr.length > 0
 * @return the largest value in array arr
 */
public int findMax()
{
    int maxVal = 0;

    for (int val : arr)
    {
        if (val > maxVal)
        {
            maxVal = val;
        }
    }

    return maxVal;
}
```

Method findMax is intended to return the largest value in the array arr. Which of the following best describes the conditions under which the method findMax will not work as intended?



**U5-10\_3**

---

- (A) The largest value in arr occurs only once and is in arr[0].
- (B) The largest value in arr occurs only once and is in arr[arr.length - 1].
- (C) The largest value in arr is negative.
- (D) The largest value in arr is zero.
- (E) The largest value in arr occurs more than once.
- 

15. Consider the following instance variable and method.

```
private int[] numbers;

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

Assume that numbers has been initialized with the following values.

{17, 34, 21, 42, 15, 69, 48, 25, 39}

Which of the following represents the order of the values in numbers as a result of the call mystery(3)?



U5-10\_3

---

- (A) {17, 20, 21, 42, 45, 69, 48, 51, 39}
- (B) {17, 20, 23, 26, 29, 32, 35, 38, 41}
- (C) {17, 37, 21, 42, 18, 69, 48, 28, 39}
- (D) {20, 23, 21, 42, 45, 69, 51, 54, 39}
- (E) {20, 34, 21, 45, 15, 69, 51, 25, 39}
- 

16. Consider the following instance variable and method. Method `wordsWithCommas` is intended to return a string containing all the words in `listOfWords` separated by commas and enclosed in braces. For example, if `listOfWords` contains `["one", "two", "three"]`, the string returned by the call `wordsWithCommas ()` should be `"{one, two, three}"`.

```
private List<String> listOfWords;

public String wordsWithCommas()
{
    String result = "{";

    int sizeOfList = /* expression */ ;

    for (int k = 0; k < sizeOfList; k++)
    {
        result = result + listOfWords.get(k);

        if ( /* condition */ )
        {
            result = result + ", ";
        }
    }

    result = result + "}";
    return result;
}
```

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



## U5-10\_3

- (A) */\* expression \*/ /\* condition \*/*  
 listOfWords.size() - 1 / k != 0
- (B) */\* expression \*/ /\* condition \*/*  
 listOfWords.size() / k != 0
- (C) */\* expression \*/ /\* condition \*/*  
 listOfWords.size() - 1 / k != sizeOfList - 1
- (D) */\* expression \*/ /\* condition \*/*  
 listOfWords.size() / k != sizeOfList - 1
- (E) */\* expression \*/ /\* condition \*/*  
 result.length() / k != 0

17. Consider the following instance variable and method.

```
private int[] arr;

/** Precondition: arr contains no duplicates;
 *         the elements in arr are in ascending order.
 * @param low an int value such that  $0 \leq \text{low} \leq \text{arr.length}$ 
 * @param high an int value such that  $\text{low} - 1 \leq \text{high} < \text{arr.length}$ 
 * @param num an int value
 */
public int mystery(int low, int high, int num)
{
    int mid = (low + high) / 2;
    if (low > high)
    {
        return low;
    }
    else if (arr[mid] < num)
    {
        return mystery(mid + 1, high, num);
    }
    else if (arr[mid] > num)
    {
        return mystery(low, mid - 1, num);
    }
    else // arr[mid] == num
    {
        return mid;
    }
}
```

What is returned by the call `mystery(0, arr.length - 1, num)`?



U5-10\_3

---

- (A) The number of elements in `arr` that are less than `num`
  - (B) The number of elements in `arr` that are less than or equal to `num`
  - (C) The number of elements in `arr` that are equal to `num`
  - (D) The number of elements in `arr` that are greater than `num`
  - (E) The index of the middle element in `arr`
- 

18. Consider the following instance variable and method.

```
private int[] numbers;

/** Precondition: numbers contains int values in no particular order.
 */
public int mystery(int num)
{
    for (int k = numbers.length - 1; k >= 0; k--)
    {
        if (numbers[k] < num)
        {
            return k;
        }
    }
    return -1;
}
```

Which of the following best describes the contents of `numbers` after the following statement has been executed?

`int m = mystery(n);`



U5-10\_3

---

- (A) All values in positions 0 through  $m$  are less than  $n$ .
- (B) All values in positions  $m+1$  through  $\text{numbers.length}-1$  are less than  $n$ .
- (C) All values in positions  $m+1$  through  $\text{numbers.length}-1$  are greater than or equal to  $n$ .
- (D) The smallest value is at position  $m$ .
- (E) The largest value that is smaller than  $n$  is at position  $m$ .
- 

19. Consider the following instance variable and method.

```
private int[] array;

/** Precondition: array.length > 0
 */
public int checkArray()
{
    int loc = array.length / 2;
    for (int k = 0; k < array.length; k++)
    {
        if (array[k] > array[loc])
        {
            loc = k;
        }
    }
    return loc;
}
```

Which of the following is the best postcondition for `checkArray` ?



**U5-10\_3**

---

- (A) Returns the index of the first element in array array whose value is greater than array[loc]
  - (B) Returns the index of the last element in array array whose value is greater than array[loc]
  - (C) Returns the largest value in array array
  - (D) Returns the index of the largest value in array array
  - (E) Returns the index of the largest value in the second half of array array
- 

20. 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?



**U5-10\_3**

---

- (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"]
-