

U5-10_5

Name _____

1. Consider the following method.

```
public ArrayList<Integer> mystery(int n)
{
    ArrayList<Integer> seq = new ArrayList<Integer>();

    for (int k = 1; k <= n; k++)
        seq.add(new Integer(k * k + 3));

    return seq;
}
```

Which of the following is printed as a result of executing the following statement?
System.out.println(mystery (6));

- (A) [3, 4, 7, 12, 19, 28]
- (B) [3, 4, 7, 12, 19, 28, 39]
- (C) [4, 7, 12, 19, 28, 39]
- (D) [39, 28, 19, 12, 7, 4]
- (E) [39, 28, 19, 12, 7, 4, 3]



U5-10_5

2. Consider the following method.

```
public static void arrayMethod(int nums[])
{
    int j = 0;
    int k = nums.length - 1;

    while (j < k)
    {
        int x = nums[j];
        nums[j] = nums[k];
        nums[k] = x;
        j++;
        k--;
    }
}
```

Which of the following describes what the method `arrayMethod()` does to the array `nums`?

- (A) The array `nums` is unchanged.
 - (B) The first value in `nums` is copied to every location in the array.
 - (C) The last value in `nums` is copied to every location in the array.
 - (D) The method generates an `ArrayIndexOutOfBoundsException`.
 - (E) The contents of the array `nums` are reversed.
-



U5-10_5

3. Consider the following methods.

```
public void changer(String x, int y)
{
    x = x + "peace";
    y = y * 2;
}

public void test()
{
    String s = "world";
    int n = 6;
    changer(s, n);

    /* End of method */
}
```

When the call `test ()` is executed, what are the values of `s` and `n` at the point indicated by */* End of method */*?

- (A) `s / n`
`world / 6`
 - (B) `s / n`
`worldpeace / 6`
 - (C) `s / n`
`world / 12`
 - (D) `s / n`
`worldpeace / 12`
 - (E) `s / n`
`peace / 12`
-



U5-10_5

4. Consider the following recursive method.

```
public static void whatsItDo(String str)
{
    int len = str.length();
    if (len > 1)
    {
        String temp = str.substring(0, len - 1);
        System.out.println(temp);
        whatsItDo(temp);
    }
}
```

What is printed as a result of the call `whatsItDo("WATCH")` ?

- (A) H
- (B) WATC
ATCH
- (C) ATC
AT
A
WATC
- (D) WAT
WA
W
WATCH
- (E) WATC
WAT
WA
-



U5-10_5

5. Consider the following recursive method.

```
/** Precondition: num ≥ 0 */
public static int what(int num)
{
    if (num < 10)
    {
        return 1;
    }
    else
    {
        return 1 + what(num / 10);
    }
}
```

Assume that `int val` has been declared and initialized with a value that satisfies the precondition of the method. Which of the following best describes the value returned by the call `what(val)` ?

- (A) The number of digits in the decimal representation of `val` is returned.
 - (B) The sum of the digits in the decimal representation of `val` is returned.
 - (C) Nothing is returned. A run-time error occurs because of infinite recursion.
 - (D) The value 1 is returned.
 - (E) The value `val/10` is returned.
-



U5-10_5

6. Consider the following recursive method.

```
public static int mystery(int n)
{
    if (n <= 1)
    {
        return 0;
    }
    else
    {
        return 1 + mystery(n / 2);
    }
}
```

Assuming that k is a nonnegative integer and $m = 2^k$, what value is returned as a result of the call `mystery(m)`?

- (A) 0
- (B) k
- (C) m
- (D) $\frac{m}{2} + 1$
- (E) $\frac{k}{2} + 1$
-



U5-10_5

7. Consider the following recursive method.

```
public static void whatsItDo(String str)
{
    int len = str.length();
    if (len > 1)
    {
        String temp = str.substring(0, len - 1);
        whatsItDo(temp);
        System.out.println(temp);
    }
}
```

What is printed as a result of the call `whatsItDo("WATCH")` ?

- WATC
- (A) WAT
WA
W
WATCH
- (B) WATC
WAT
WA
W
WA
WATC
W
WA
WAT
WATC
W
WA
WATCH
WATCH
WATC
WAT
WA
(E) W
WA
WAT
WATC
WATCH



U5-10_5

8. Consider the following recursive method.

```
public int recur(int n)
{
    if (n <= 10)
        return n * 2;
    else
        return recur(recur(n / 3));
}
```

What value is returned as a result of the call `recur(27)`?

- (A) 8
- (B) 9
- (C) 12
- (D) 16
- (E) 18
-



U5-10_5

9. Consider the following two classes.

```
public class Dog
{
    public void act()
    {
        System.out.print("run ");
        eat();
    }
    public void eat()
    {
        System.out.print("eat ");
    }
}
public class UnderDog extends Dog
{
    public void act()
    {
        super.act();
        System.out.print("sleep ");
    }
    public void eat()
    {
        super.eat();
        System.out.print("bark ");
    }
}
```

Assume that the following declaration appears in a class other than Dog.

Dog fido = new UnderDog ();

What is printed as a result of the call fido.act() ?



U5-10_5

- ☐ (A) run eat
- ☐ (B) run eat sleep
- ☐ (C) run eat sleep bark
- ☐ (D) run eat bark sleep
- ☐ (E) Nothing is printed due to infinite recursion.
-

10. Consider the following two classes.

```
public class A
{
    public void show()
    {
        System.out.print("A");
    }
}
```

```
public class B extends A
{
    public void show()
    {
        System.out.print("B");
    }
}
```

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

```
A obj = new B();
obj.show();
```



U5-10_5

- ☐ A
- ☐ B
- ☐ C AB
- ☐ D BA
- ☐ E The code results in a runtime error.
-

11. Consider the following two methods, which appear within a single class.

```
public static void changeIt(int[] arr, int val, String word)
{
    arr = new int[5];
    val = 0;
    word = word.substring(0, 5);

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

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

    changeIt(nums, value, name);

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

    System.out.print(value + " ");
    System.out.print(name);
}
```

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



U5-10_5

- (A) 0 0 0 0 0 0 black
- (B) 0 0 0 0 0 6 blackboard
- (C) 1 2 3 4 5 6 black
- (D) 1 2 3 4 5 0 black
- (E) 1 2 3 4 5 6 blackboard
-



U5-10_5

12. Consider the following Util class, which contains two methods. The completed sum1D method returns the sum of all the elements of the 1-dimensional array a. The incomplete sum2D method is intended to return the sum of all the elements of the 2-dimensional array m.

```
public class Util
{
    /** Returns the sum of the elements of the 1-dimensional array a */
    public static int sum1D(int[] a)
    { /* implementation not shown */ }

    /** Returns the sum of the elements of the 2-dimensional array m */
    public static int sum2D(int[][] m)
    {
        int sum = 0;

        /* missing code */

        return sum;
    }
}
```

Assume that sum1D works correctly. Which of the following can replace */* missing code */* so that the sum2D method works correctly?

- I.

```
for (int k = 0; k < m.length; k++)
{
    sum += sum1D(m[k]);
}
```
- II.

```
for (int[] row : m)
{
    sum += sum1D(row);
}
```
- III.

```
for (int[] row : m)
{
    for (int v : row)
    {
        sum += v;
    }
}
```



U5-10_5

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



U5-10_5

13. Consider the problem of finding the maximum value in an array of integers. The following code segments are proposed solutions to the problem. Assume that the variable `arr` has been defined as an array of `int` values and has been initialized with one or more values.

```
I.  int max = Integer.MIN_VALUE;
    for (int value : arr)
    {
        if (max < value)
        {
            max = value;
        }
    }
```

```
II. int max = 0;
    boolean first = true;
    for (int value : arr)
    {
        if (first)
        {
            max = value;
            first = false;
        }
        else if (max < value)
        {
            max = value;
        }
    }
```

```
III. int max = arr[0];
     for (int k = 1; k < arr.length; k++)
     {
         if (max < arr[k])
         {
             max = arr[k];
         }
     }
```

Which of the code segments will always correctly assign the maximum element of the array to the variable `max` ?

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



U5-10_5

14. In the following code segment, assume that the ArrayList `wordList` has been initialized to contain the String values ["apple", "banana", "coconut", "lemon", "orange", "pear"].

```
int count = 0;
for (String word : wordList)
{
    if (word.indexOf("a") >= 0)
    {
        count++;
    }
}
System.out.println(count);
```

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

- (A) 1
 - (B) 2
 - (C) 3
 - (D) 4
 - (E) 5
-



U5-10_5

15. Consider the definition of the Person class below. The class uses the instance variable `adult` to indicate whether a person is an adult or not.

```
public class Person
{
    private String name;
    private int age;
    private boolean adult;
    public Person (String n, int a)
    {
        name = n;
        age = a;
        if (age >= 18)
        {
            adult = true;
        }
        else
        {
            adult = false;
        }
    }
}
```

Which of the following statements will create a Person object that represents an adult person?

- (A) `Person p = new Person ("Homer", "adult");`
 - (B) `Person p = new Person ("Homer", 23);`
 - (C) `Person p = new Person ("Homer", "23");`
 - (D) `Person p = new Person ("Homer", true);`
 - (E) `Person p = new Person ("Homer", 17);`
-



U5-10_5

16. Consider the following class definition. Each object of the class `Item` will store the item's name as `itemName`, the item's regular price, in dollars, as `regPrice`, and the discount that is applied to the regular price when the item is on sale as `discountPercent`. For example, a discount of 15% is stored in `discountPercent` as 0.15.

```
public class Item
{
    private String itemName;
    private double regPrice;
    private double discountPercent;
    public Item (String name, double price, double discount)
    {
        itemName = name;
        regPrice = price;
        discountPercent = discount;
    }
    public Item (String name, double price)
    {
        itemName = name;
        regPrice = price;
        discountPercent = 0.25;
    }
    /* Other methods not shown */
}
```

Which of the following code segments, found in a class other than `Item`, can be used to create an item with a regular price of \$10 and a discount of 25% ?

1. `Item b = new Item("blanket", 10.0, 0.25);`
2. `Item b = new Item("blanket", 10.0);`
3. `Item b = new Item("blanket", 0.25, 10.0);`



U5-10_5

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

17. Consider the following method.

```
public void changeIt(int[] arr, int index, int newValue)
{
    arr[index] += newValue;
}
```

Which of the following code segments, if located in a method in the same class as `changeIt`, will cause the array `myArray` to contain `{0, 5, 0, 0}` ?

- ☐ (A) `int[] myArray = new int[4];`
`changeIt(myArray, 1, 5);`
- ☐ (B) `int[] myArray = new int[4];`
`changeIt(myArray, 2, 5);`
- ☐ (C) `int[] myArray = new int[4];`
`changeIt(myArray, 5, 1);`
- ☐ (D) `int[] myArray = new int[5];`
`changeIt(myArray, 1, 4);`
- ☐ (E) `int[] myArray = new int[5];`
`changeIt(myArray, 1, 5);`
-



U5-10_5

- 18.** Consider the following class definitions.

```
public class Book
{
    private String bookTitle;
    public Book()
    {
        bookTitle = "";
    }
    public Book(String title)
    {
        bookTitle = title;
    }
}

public class TextBook extends Book
{
    private String subject;
    public TextBook(String theSubject)
    {
        subject = theSubject;
    }
}
```

The following code segment appears in a method in a class other than Book or TextBook.

```
Book b = new TextBook("Psychology");
```

Which of the following best describes the effect of executing the code segment?



U5-10_5

- The `TextBook` constructor initializes the instance variable `subject` with the value of the parameter `theSubject`, and then invokes the zero-parameter `Book` constructor, which initializes the instance variable `bookTitle` to `""`.
- The `TextBook` constructor initializes the instance variable `subject` with the value of the parameter `theSubject`, and then invokes the one-parameter `Book` constructor with `theSubject` as the parameter, which initializes the instance variable `bookTitle` to the value of the parameter `theSubject`.
- There is an implicit call to the zero-parameter `Book` constructor. The instance variable `bookTitle` is then initialized to `""`. Then, the instance variable `subject` is initialized with the value of the parameter `theSubject`.
- The code segment will not execute because the `TextBook` constructor does not contain an explicit call to one of the `Book` constructors.
- The code segment will not execute because the `TextBook` constructor does not have a parameter for the title of the book.
-

19. Consider the following code segment, which is intended to declare and initialize the two-dimensional (2D) `String` array `things`.

```
/* missing code */ = {{"spices", "garlic", "onion", "pepper"},  
{"clothing", "hat", "scarf", "gloves"},  
{"plants", "tree", "bush", "flower"},  
{"vehicles", "car", "boat", "airplane"}};
```

Which of the following could replace `/* missing code */` so that `things` is properly declared?

- ☐ A `new String[][] things`
- ☐ B `new(String[][]) things`
- ☐ C `String[] String[] things`
- ☐ D `String[][] things`
- ☐ E `[][]String things`
-



U5-10_5

20. Consider the following class, which uses the instance variable `balance` to represent a bank account balance.

```
public class BankAccount
{
    private double balance;
    public double deposit(double amount)
    {
        /* missing code */
    }
}
```

The `deposit` method is intended to increase the account balance by the deposit amount and then return the updated balance. Which of the following code segments should replace `/* missing code */` so that the `deposit` method will work as intended?

- (A) `amount = balance + amount;`
`return amount;`
 - (B) `balance = amount;`
`return amount;`
 - (C) `balance = amount;`
`return balance;`
 - (D) `balance = balance + amount;`
`return amount;`
 - (E) `balance = balance + amount;`
`return balance;`
-