

University of Washington, Tacoma
Institute of Technology
305 Programming Assessment Exam

Instructions: Please *circle* your answers for the multiple-choice questions and write the answers to the programming questions in the space provided on the same page as the question.

1. Given the following array declaration:

```
int[] powersOf2 = new int[6];
```

Which of the following pieces of code correctly stores the first six powers of 2 (namely 1, 2, 4, 8, 16, 32) into the elements of the array named `powersOf2`?

I.

```
powersOf2[0] = 1;
for (int i = 1; i < powersOf2.length; i++)
    powersOf2[i] = 2 * powersOf2[i - 1];
```

II.

```
for (int i = 0; i < powersOf2.length; i++) {
    powersOf2[i] = 1;
    for (int j = 0; j < i; j++)
        powersOf2[i] += powersOf2[j];
}
```

III.

```
for (int i = 1; i < powersOf2.length; i++)
    powersOf2[i] = Math.pow(i, 2);
```

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

2. Which statement about `ArrayLists` is NOT true?

- A. An `ArrayList` stores elements with 0-based indexes.
- B. Arrays and `ArrayLists` are interchangeable; a method that accepts an array as a parameter can be passed an `ArrayList` instead.
- C. An `ArrayList` keeps track of its own size.
- D. The `ArrayList` method `add(Element e)` stores elements in the order in which they were added.
- E. It requires a constant amount of time to access any individual element in an `ArrayList`.

3. Consider the following method:

```
public String encrypt(String word) {
    int pos = word.length() / 2;
    if (pos >= 1) {
        return encrypt(word.substring(pos)) + encrypt(word.substring(0, pos));
    } else {
        return word;
    }
}
```

What string is returned from `encrypt("SECRET")` ?

- A. TSECRE
- B. RETSEC
- C. TERCES
- D. CESTER
- E. ETRECS

4. Consider the following interface and class:

```
public interface I {
    void m1();
    void m2();
}

public class C implements I {
    code for class C;
}
```

Assuming that the code is valid and compiles, which of the following statements must be true?

- A. Class C inherits methods `m1` and `m2` from I, and therefore it is illegal for the code for class C to write an `m1` or `m2` method.
- B. The code for class C may not declare an additional method as follows:

```
void m1(int i);
```
- C. The code for class C is not allowed to have any constructors.
- D. The code for class C must contain implementations of methods `m1` and `m2`.
- E. Class C also acts as an interface, and other classes may choose to implement C.

The next two questions refer to the following linked list node class:

```
public class ListNode {
    private Object data;
    private ListNode next;

    public ListNode(Object initialData, ListNode initialNext) {
        setData(initialData);
        setNext(initialNext);
    }

    public Object getData() {
        return data;
    }

    public ListNode getNext() {
        return next;
    }

    public final void setData(Object newData) {
        data = newData;
    }

    public final void setNext(ListNode newNext) {
        next = newNext;
    }
}
```

5. If head refers to the first node of a linked list with two nodes, as follows:

```
ListNode head = new ListNode(new Integer(1), new ListNode(new Integer(2), null));
```

head → 1 → 2

Which of the following statements would change the list to the following?

head → 3 → 1 → 2

- A. head.setNext(new ListNode(new Integer(3), head.getNext()));
- B. head.setNext(new ListNode(new Integer(3), head));
- C. head = new ListNode(new Integer(3), head);
- D. head.setNext(new ListNode(new Integer(3), head));
- E. head = new ListNode(new Integer(3));

6. Consider the following method:

```
public ListNode mince(ListNode head) {
    ListNode r = null;
    ListNode p = null;

    while (head != null) {
        r = head.getNext();
        head.setNext(p);
        p = head;
        head = r;
    }

    return p;
}
```

If head refers to the first node of a linked list with five nodes, as follows:

```
ListNode head = new ListNode("A", new ListNode("B",
    new ListNode("C", new ListNode("D", new ListNode("E", null))));
```

head → A → B → C → D → E

Which of the following lists is returned by the following?

```
head = mince(head);
```

- A. head → B → C → D → E
- B. head → E
- C. head → A → B → C → D
- D. head → A → B → C → D → E
- E. head → E → D → C → B → A

7. Consider the following method:

```
public void sentence(String major, String fred, String foo) {  
    System.out.println("Many a " + foo + " in the " + fred + " of " + major);  
}
```

What output is produced from the following code?

```
String major = "fred";  
String fred = "computer";  
String honor = "grade";  
String department = "student";  
  
sentence(fred, "honor", department);
```

- A. Many a foo in the fred of major
- B. Many a student in the honor of computer
- C. Many a department in the honor of fred
- D. Many a department in the grade of computer
- E. Many a fred in the honor of department

8. What output is produced by the following code?

```
ArrayList<Integer> list = new ArrayList<Integer>();  
for (int i = 1; i <= 8; i++)  
    list.add(new Integer(i));  
for (int i = 0; i < list.size(); i++)  
    list.remove(i); // remove(i) removes the element at the specified position  
for (int i = 0; i < list.size(); i++)  
    System.out.print(list.get(i) + " ");
```

- A. 2 4 6 8
- B. 1 3 5 7
- C. 1 2 3 4 5 6 7 8
- D. IndexOutOfBoundsException
- E. The code does not produce any output.

The next two questions refer to the following classes:

```
public class Mammal extends Animal {
    public void method3() {
        System.out.print("Mammal 3 ");
        method2();
    }
    public void method4(String s) {
        System.out.print("Mammal 4 ");
    }
}

public class Tiger extends Mammal {
    public void method1() {
        System.out.print("Tiger 1 ");
        super.method2();
    }
    public void method2() {
        System.out.print("Tiger 2 ");
    }
}

public class Animal {
    public void method1() {
        System.out.print("Animal 1 ");
    }
    public void method2() {
        System.out.print("Animal 2 ");
    }
    public void method3() {
        System.out.print("Animal 3 ");
    }
    public void method4(Object o) {
        System.out.print("Animal 4");
    }
}
```

9. Which of the following is NOT a legal statement?

- A. `Animal a = new Mammal();`
- B. `Animal a = new Tiger();`
- C. `Mammal m = new Tiger();`
- D. `Tiger t = new Tiger();`
- E. `Mammal m = new Animal();`

10. What output is produced by the following code fragment?

```
Animal[] animals = {new Tiger(), new Animal(), new Mammal()};
for (Animal a : animals) {
    a.method1();
    System.out.println();
    a.method2();
    System.out.println();
    a.method3();
    System.out.println();
    a.method4("method4");
    System.out.println("\n");
}
Mammal m = new Mammal();
m.method4("method4");
System.out.println();
Tiger t = new Tiger();
t.method4("method4");
```

This question refers to the `ListNode` class used in multiple-choice problems 5 and 6.

P1.

Write a method named `countDuplicates` that takes a parameter of type `ListNode` and returns the number of elements that repeat an earlier value. You may assume that the list elements are ordered such that all duplicates occur consecutively.

For example, if `head` refers to the front of a list containing the following values:

`head → 4 → 3 → 3 → 9 → 9 → 9 → 7 → 11 → 17 → 17 → 17 → 17`

The following method call:

```
countDuplicates(head)
```

would return 6, because there is 1 duplicate of 3, 2 duplicates of 9, and 3 duplicates of 17. If the list is empty or has no duplicate elements, you should return 0. You must preserve the original order and value of the elements of the list. Your method should work for any type of objects in the list, not just `Integer`s.

P2.

Write a recursive method named `writeChars` that accepts an integer n as a parameter and writes out n characters as follows. If you are asked to write out an odd number of characters, the middle character of the output should be an asterisk (*). If you are asked to write out an even number of characters, the two middle characters must be asterisks. Before the asterisk(s) you should write out less-than characters (<). After the asterisk(s) you should write out greater-than characters (>). You should write all of the characters on a single line of output.

For example, the following code:

```
writeChars(5);
System.out.println(); // to complete the line of output
writeChars(12);
System.out.println(); // to complete the line of output
```

Would produce the following output:

```
<<*>>
<<<<<**>>>>>
```

If your method is passed 1 or 2 as an argument, it should write out just asterisks. You may assume that it is never passed a value less than 1. ***You may not use a loop to solve this problem; you MUST use recursion.***