

University of Washington
Computer Science & Engineering 142 (Computer Programming I), Summer 2005
Sample Final Exam

1. Expressions, 10 points. For each expression in the left-hand column, indicate its value in the right-hand column. Be sure to list a constant of appropriate type (e.g., 7.0 rather than 7 for a double, Strings in quotes).

```
String[] words = {"hello", "how", "are", "you"};
Point p = new Point(5, 3);
```

<u>Expression</u>	<u>Value</u>
13 / 2 * 3 % 5 - 1	_____
2 >= 1 + 3 !(p.getX() < p.getY())	_____
30 == (6 * words.length + 1)	_____
13 % 5 + 5 * Math.min(p.getX(), p.getY()) / 4	_____
words[1] + 3 * 4 + words[2] + 1 + 2	_____

2. Array Mystery, 15 points. Write the output produced by the following program.

```
public static void mystery(int[] list) {
    for (int i = 2; i < list.length; i++) {
        list[i] = list[i] + list[i - 1] + list[i - 2];
    }
}
```

For each call below, indicate what value is returned:

Method Call	Value Returned
int[] a1 = {8}; mystery(a1);	_____
int[] a2 = {2, 7, 12}; mystery(a2);	_____
int[] a3 = {3, 0, 1, 4, 7}; mystery(a3);	_____
int[] a4 = {0, 1, 2, 3, 4, 5}; mystery(a4);	_____
int[] a5 = {7, 4, -10, 8, 2}; mystery(a5);	_____

3. Inheritance and polymorphism, 15 points. Given the following inheritance hierarchy of classes:

```
public class Mammal extends SeaCreature {
    public void method1() {
        System.out.println("warm-blooded");
    }
}

public class SeaCreature {
    public void method1() {
        System.out.println("creature 1");
    }

    public void method2() {
        System.out.println("creature 2");
    }

    public String toString() {
        return "ocean-dwelling";
    }
}

public class Whale extends Mammal {
    public void method1() {
        System.out.println("spout");
    }

    public String toString() {
        return "BIG!";
    }
}

public class Squid extends SeaCreature {
    public void method2() {
        System.out.println("tentacles");
    }

    public String toString() {
        return "squid";
    }
}
```

Consider the following code fragment. What output is produced by this code?

```
SeaCreature[] elements = {new Squid(), new Whale(), new SeaCreature(),
                          new Mammal()};
for (int i = 0; i < elements.length; i++) {
    System.out.println(elements[i]);
    elements[i].method1();
    elements[i].method2();
    System.out.println();
}
```

4. File Processing, 15 points. Write a static method named `processData` that takes as a parameter a `Scanner` holding a sequence of integers and that reports cumulative sums of each line's sequence of integers, along with the average of the numbers on that line. For example, if the `Scanner` contains the following data:

```
8 3 2 5 7
13 5 9 -2
2 3
```

Your method should produce the following output:

```
Sum of 1 = 8
Sum of 2 = 11
Sum of 3 = 13
Sum of 4 = 18
Sum of 5 = 25
Average = 5.0

Sum of 1 = 13
Sum of 2 = 18
Sum of 3 = 27
Sum of 4 = 25
Average = 6.25

Sum of 1 = 2
Sum of 2 = 5
Average = 2.5
```

You are to exactly reproduce the format of these sample outputs. You may assume that each line has at least one integer to be processed.

5. Arrays, 15 points. Write a static method `printNumber` that takes an array of integers that represent the digits of a large integer and that prints the integer with commas between every 3 digits starting from the right. Each array element will store one digit of the number with array element 0 storing the first digit of the number, array element 1 storing the second digit of the number and so on. You may assume that the values in the array are all between 0 and 9 and that there are no leading 0's. For example, if the array contains the digits {1, 5, 0, 0}, then your method should produce the following output:

1,500

If the array instead contains {3, 8, 4, 9, 2, 1, 4, 7}, then your method should produce the following output:

38,492,147

Your method might not print any commas if the number is small enough. For example, if the array contains the digits {2, 5, 0}, then your method should produce the following output:

250

Your method should produce an entire line of output so that if it is called several times in a row, each call will produce a separate line of output.

6. Arrays, 15 points. Write a static method `interleave` that takes two arrays of integers as parameters and that returns a new array that contains the result of interleaving the elements of the two arrays. It should not alter either of its parameters. Two arrays are interleaved by taking elements in the following order:

```
1st element of 1st array
1st element of 2nd array
2nd element of 1st array
2nd element of 2nd array
3rd element of 1st array
3rd element of 2nd array
```

... and so on. For example, if the variables `array1` and `array2` contain the following values:

```
int[] array1 = {1, 8, 3, 9};
int[] array2 = {2, 12, 6, 14};
```

Then the call `interleave(array1, array2)` should return a new array that stores the following values:

```
{1, 2, 8, 12, 3, 6, 9, 14}
```

The order of the parameters matters. For example, `interleave(array2, array1)` should produce the following array as its result:

```
{2, 1, 12, 8, 6, 3, 14, 9}
```

One array might be longer than the other, in which case any extra values from the longer array should simply be appended at the end of the result. For example, given the following arrays:

```
int[] array1 = {1, 8, 3, 9};
int[] array2 = {82, 7, 4, 2, 1, 6, 5, 0, 18};
```

Then the call `interleave(array1, array2)` should produce the following result:

```
{1, 82, 8, 7, 3, 4, 9, 2, 1, 6, 5, 0, 18}
```

Notice that the first four values of the two arrays have been interleaved and the excess values from the second array `{1, 6, 5, 0, 18}` have been included at the end of the result. In this case the second array was longer, but it could be the case that the first array is longer. Either array could also be empty, in which case the result should contain the values from the other array. Notice that your method is to return a new array.

- 7 Defining new types of objects, 15 points. Consider that you are writing a new type of objects named `LineSegment`. A line segment is a line between two points (x_1, y_1) and (x_2, y_2) . The following already exists:

```
// Represents a line segment between two Points.
public class LineSegment {
    private Point p1;
    private Point p2;

    // Constructs a new Line that contains the given two Points.
    public LineSegment(Point p1, Point p2) {
        this.p1 = p1;
        this.p2 = p2;
    }

    // Returns whether the given other object o is a LineSegment
    // with the same endpoints as this LineSegment.
    public boolean equals(Object o)

    public Point getP1()
    public Point getP2()
}
```

Add the following to the `LineSegment` class:

- A constructor that accepts four integers $x_1, y_1, x_2,$ and y_2 as parameters, rather than two `Point` objects.
- A method named `getSlope` that returns the slope of the current line segment. Recall that the slope of a line can be found by computing its change-in-y divided by its change-in-x.

If the line segment is a vertical line, it has an undefined slope. Handle this case by throwing an `IllegalArgumentException`.