

University of Washington

Computer Science & Engineering 142 (Computer Programming I), Summer 2005

Sample Final Exam Key

1. Expressions

Expression	Value
<code>13 / 2 * 3 % 5 - 1</code>	2
<code>2 >= 1 + 3 !(p.getX() < p.getY())</code>	true
<code>30 == (6 * words.length + 1)</code>	false
<code>13 % 5 + 5 * Math.min(p.getX(), p.getY()) / 4</code>	6.75
<code>words[1] + 3 * 4 + words[2] + 1 + 2</code>	"how12are12"

2. Array Mystery

Method Call	Resulting Array
<code>mystery(a1)</code>	{8}
<code>mystery(a2)</code>	{2, 7, 21}
<code>mystery(a3)</code>	{3, 0, 4, 8, 19}
<code>mystery(a4)</code>	{0, 1, 3, 7, 14, 26}
<code>mystery(a5)</code>	{7, 4, 1, 13, 16}

3. Inheritance and Polymorphism

```
squid
creature 1
tentacles
```

```
BIG!
spout
creature 2
```

```
ocean-dwelling
creature 1
creature 2
```

```
ocean-dwelling
warm-blooded
creature 2
```

4. File Processing

```
public static void processData(Scanner input) {
    while (input.hasNextLine()) {
        String line = input.nextLine();
        Scanner lineScan = new Scanner(line);

        int sum = 0;
        int count = 0;
        while (lineScan.hasNextInt()) {
            sum += lineScan.nextInt();
            count++;
            System.out.println("Sum of " + count + " = " + sum);
        }

        double average = (double) sum / count;
        System.out.println("Average = " + average);
        System.out.println();
    }
}
```

5. Arrays

```
public static void printNumber(int[] digits) {
    for (int i = 0; i < digits.length; i++) {
        if (i > 0 && (digits.length - i) % 3 == 0)
            System.out.print(",");
        System.out.print(digits[i]);
    }
    System.out.println();
}
```

6. Arrays

```
public static int[] interleave(int[] list1, int[] list2) {
    int[] result = new int[list1.length + list2.length];
    int min = Math.min(list1.length, list2.length);
    for (int i = 0; i < min; i++) {
        result[2 * i] = list1[i];
        result[2 * i + 1] = list2[i];
    }
    for (int i = min; i < list1.length; i++)
        result[i + min] = list1[i];
    for (int i = min; i < list2.length; i++)
        result[i + min] = list2[i];
    return result;
}
```

7. Defining new types of objects (two solutions to constructor shown)

```
public LineSegment(int x1, int y1, int x2, int y2) {
    this.p1 = new Point(x1, y1);
    this.p2 = new Point(x2, y2);
}

// alternate solution
public LineSegment(int x1, int y1, int x2, int y2) {
    this(new Point(x1, y1), new Point(x2, y2));
}

public double getSlope() {
    if (p1.getX() == p2.getX()) {
        throw new IllegalArgumentException();
    }

    return (double) (p2.getY() - p1.getY()) / (p2.getX() - p1.getX());
}
```