

Introduction to Computer Science I
Spring 2010
SAMPLE MID-TERM EXAM

You have **50 minutes** to complete this 100 point exam.

1. (15 points) Consider the code fragment below. **Mark each location** where an *automatic cast* will occur. Also find each location where an *explicit cast* must be inserted for the code to compile successfully, and **correct the line** with that explicit cast included. [Note: Be aware that some lines may require more than one cast!]

```
byte b = 13;
int i = b;
short s = i;
int i2 = s + i;
short s2 = s + 3;
boolean b2 = (i < s);
```

2. (15 points) Consider the Java code below and answer the questions that follow.

```
System.out.print("Enter a value for a: ");
int a = keyboard.nextInt();
System.out.print("Enter a value for b: ");
int b = keyboard.nextInt();

if (a < 0) {
    a = -a;
}

int i = 0;
while (i < a) {

    int k = 0;
    int j = 100;
    while (j >= b) {
        k = j * 2;
        System.out.print(k);
        System.out.print('-');
        System.out.println(j);
        j = j - 1;
    }

    k = k + 1;
    i = i + 1;

}
```

- (a) What is the output of this code if the user enters 2 for **a** and 98 for **b**?
- (b) What is the output of this code if the user enters -3 for **a** and 100 for **b**?

3. (10 points) The following method contains an error that will prevent it from compiling. **Find and correct it.**

```
public static int quux () {

    int x = 0;
    while (x <= 0) {

        System.out.print("Enter a value: ");
        x = keyboard.nextInt();
        if (x > 100) {
            int y = x;
        } else {
            y = 2 * x;
        }

    }

    return y;

}
```

4. (30 points) **Write a method** named `printTable` that accepts a *height* and a *width* as parameters and then prints a table of those dimensions, where the table follows the following pattern:

```
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
```

The above table is the result of calling `printTable(4, 5)`.

5. (30 points) We want to write a program that allows the user to enter a list of positive numbers and then prints out the *mean* number from that list. That is, if the user enters the values 7, 4, 3.5, and 11.5, then the mean is $\frac{7+4+3.5+11.5}{4} = \frac{26}{4} = 6.5$. The main method of this program is:

```
public static void main (String[] args) {

    double x = getMeanFromUser();
    System.out.println("Max = " + x);

}
```

Write the method `getMeanFromUser`. The user should be able to enter an arbitrary number of positive values. As soon as the user enters a non-positive value, the program should accept that value as an indication that the user has no more values to enter. After the user has entered all of her numbers, the method should return the mean value entered. Do not worry about a user that enters non-numeric values.