

CMSC131
Fall 2004
Midterm #1

Grader Use Only:

#1		(35)
#2		(20)
#3		(20)
#4		(25)
Total		(100)

First Name: _____

Last Name: _____

Student ID: _____

Section time (10am/11am) _____ **TAs:** _____

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature: _____

General Rules:

- This exam is closed-book and closed-notes.
- Write your answers in the space provided. If you need additional space, raise your hand.
- If you have a question, please raise your hand.
- Total point value is 100 points. Point values attached to individual problems are subject to possible (minor) changes.
- Good luck!

Problem 1 (35 pts)

- a. (4 pts) Which of the following are valid Java identifiers? (Circle the valid ones.)

Car \$chultzie per%cent _22Skidoo

- b. (2 pts) Are the following two identifiers considered the same? Yes / No. (Circle one)

temperature TEMPERATURE

- c. (2 pts) What value is assigned to a reference variable to indicate that it refers to **NO** object? (Circle one.)

i. null ii. 0 (zero) iii. "" (empty string) iv. JOptionPane.NO_OBJECT

- d. (2 pts) How many distinct String object instances are created in the following code segment?

```
String s1 = new String("Hello");  
String s2 = "GoodBye";  
String s3 = s1;
```

Answer: _____

- e. (2 pts) (Fill in the blank) The special area of memory where object instances are created and

stored is called the _____

- f. (5 pts) Which of the following are **NOT** Java primitive types? (Select all that apply.)

i. integer ii. exponential iii. int iv. Float v. String

- g. (2 pts) As described in lecture, the software development lifecycle has five phases (parts). Name **ANY TWO** of them. (You do not need to describe them.)

- h. (4 pts) Give a single Java statement (using **System.out.println**) that outputs the following line:

Bob said, "It's C:\WINDOW"

- i. (4 pts) For each of the following declaration/initialization statements, indicate whether it is valid or invalid according to Java's rules on assigning types. (Circle the correct answer in each case.)

`int i = 4L;` VALID / INVALID

`boolean b = 0;` VALID / INVALID

`float f = 1.75;` VALID / INVALID

`double d = 2.0E-4;` VALID / INVALID

- j. (6 pts) **Add parentheses** to specify completely the order in which the following expressions will be evaluated. You **do not** need to specify the final value of the expressions. The variables associated with the expressions are:

```
int w = 10;
int x = 20;
int y = 30;
boolean b;
```

i. `b = w < x && y < 30 ;`

ii. `x = w *= y ++ - 2 ;`

iii. `y = - x + - y * 5 % 3 ;`

Note: Add parentheses only.
DO NOT evaluate.

- k. (2 pts) What will occur if the following code segment is executed? Briefly explain (in one or two sentences).

```
String s = "Summer";
s = null;
System.out.println( s.length() );
```

Answer:

Problem 2 (20 pts)

Show the output produced by each of the following program segments. **(Be careful.** We have hidden some traps!)

a. (10 pts)

```
int x = 1;
int y = 1;
int w = 10;
int s = 21;

if( (x == 1) || ( y++ > 1) )
    x++;
System.out.println(x);
System.out.println(y);

System.out.println(w += 2);
System.out.println(s % 10);
if (s > 30 && true)
    if (s / 2 > 10)
        System.out.println("One");
else
    System.out.println("Two");
    System.out.println("Three");
```

b. (10 pts)

```
int x = 7;
while (x > 0) {
    System.out.println(x);
    x -= 2;
}
System.out.println(x);
```

Problem 3 (20 pts)

For each of the following parts, give a code fragment to achieve the given task. In each case we have given variable declarations, but we have intentionally not given the values of the variables. You may declare additional variables if needed.

- a. (5 pts) You are given three variables **x**, **y**, **z**, declared as shown below. Give a code fragment that does the following. If **x** is negative (less than 0), it assigns **z** to be the sum of **x** and **y**. Otherwise, it assigns **z** to be the maximum of **x** and **y**. You may **not** use any methods from the Java class library

```
long x = ...;
int y = ...;
double z;
/* Write your code fragment below */
```

- b. (5 pts) You are given two variables **d** and **e** below, each of type **int**, and the third variable **b** of type **boolean**. Give a code fragment that sets **b** to **true** if **d** and **e** are both even or both odd, and sets **b** to **false** otherwise.

```
int d = ...;
int e = ...;
boolean b;
/* Write your code fragment below */
```

- c. (5 pts) You are given a variable **q** of type **float**. Give a code fragment that modifies **q** so that it is truncated to only two decimal places. For example, if **q** had been 123.6789, after your fragment is finished, **q** will have the value 123.67. You may **not** use any methods from the Java class library. (**Hint**: First multiply **q** by 100.)

```
float q = ...;  
/* Write your code fragment below */
```

- d. (5 pts) You are given four variables, **a**, **b**, **c**, and **max**, all of type **float**. Give a code fragment that assigns **max** to be the **maximum** of **a**, **b**, and **c**. For example, if these variables contained the values 17, 2, and -92, then **max** would be set to 17. You may **not** use any methods from the Java class library.

```
float a = ...;  
float b = ...;  
float c = ...;  
float max;  
/* Write your code fragment below */
```

Problem 4 (25 pts)

Write a complete Java program that computes the numeric average of a set of scores. Your program will read the number (as an integer) of scores to process, followed by the actual scores (as doubles), and will print the numeric average of these scores. For example, if we wanted to compute the numeric average of

50.5 49.5 40.0 60.0

we would first input the value 4 (number of scores), then input each of the scores. The output of the program is a message that uses the following format:

Average: <numericAverage>

where <numericAverage> denotes the computed average.

For the above example the output would be:

Average: 50.0

The following specifications apply to this problem:

- Use the prompt string “**Enter Number**” to read the number of scores.
- Use the prompt string “**Enter Value**” to read each score value.
- Use **JOptionPane** method for input and output operations.
- Each number will be read using a separate call to **JOptionPane**.
- The name of the class will be **ComputeAvg**.
- You may assume that all inputs are **valid**, and the number of scores is 1 or greater.
- Meaningful variable names are NOT required. Try to maintain good indentation.

WRITE YOUR PROGRAM ON THE NEXT PAGE.

WRITE YOUR SOLUTION TO PROBLEM 3 ON THIS PAGE.