

CMSC131

Summer 2004

Midterm #1

Name: _____

Student ID: _____

Section: _____

Grader Use Only:

#1		(15)
#2		(16)
#3		(35)
#4		(34)
Total		(100)

Problem 1 (15 points)

a. (3 pts) Which of the following are considered valid identifiers? Circle the valid ones.

altitude Robot 44Value

b. (2 pts) Mention two kinds of local variables we discussed in class.

c. (1 pt) What is the name of the memory area where objects are created?

d. (2 pts) Mention two kinds of methods we discussed in class.

e. (1 pt) How many objects are associated with the following code segment?

String m = null;

f. (1 pt) In Java, which method is used to start the execution of a program?

g. (2 pts) Mention two primitive types discussed in class.

h. (2 pts) Write the file name extension associated with a file containing Java source code.

i. (1 pt) What is the fundamental difference between a do-while and while statement?

Problem 2 (16 points)

a. (8 pts) Write the output generated by the following code segment.

```
double valueOne = 20.0, valueTwo = 30.0;

if ((valueOne > valueTwo) || (valueOne >= 0.0)) {
    valueOne -= 1.0;
    System.out.println("valueOne: " + valueOne);
} else {
    valueTwo += 2.0;
    System.out.println("valueTwo: " + valueTwo);
}

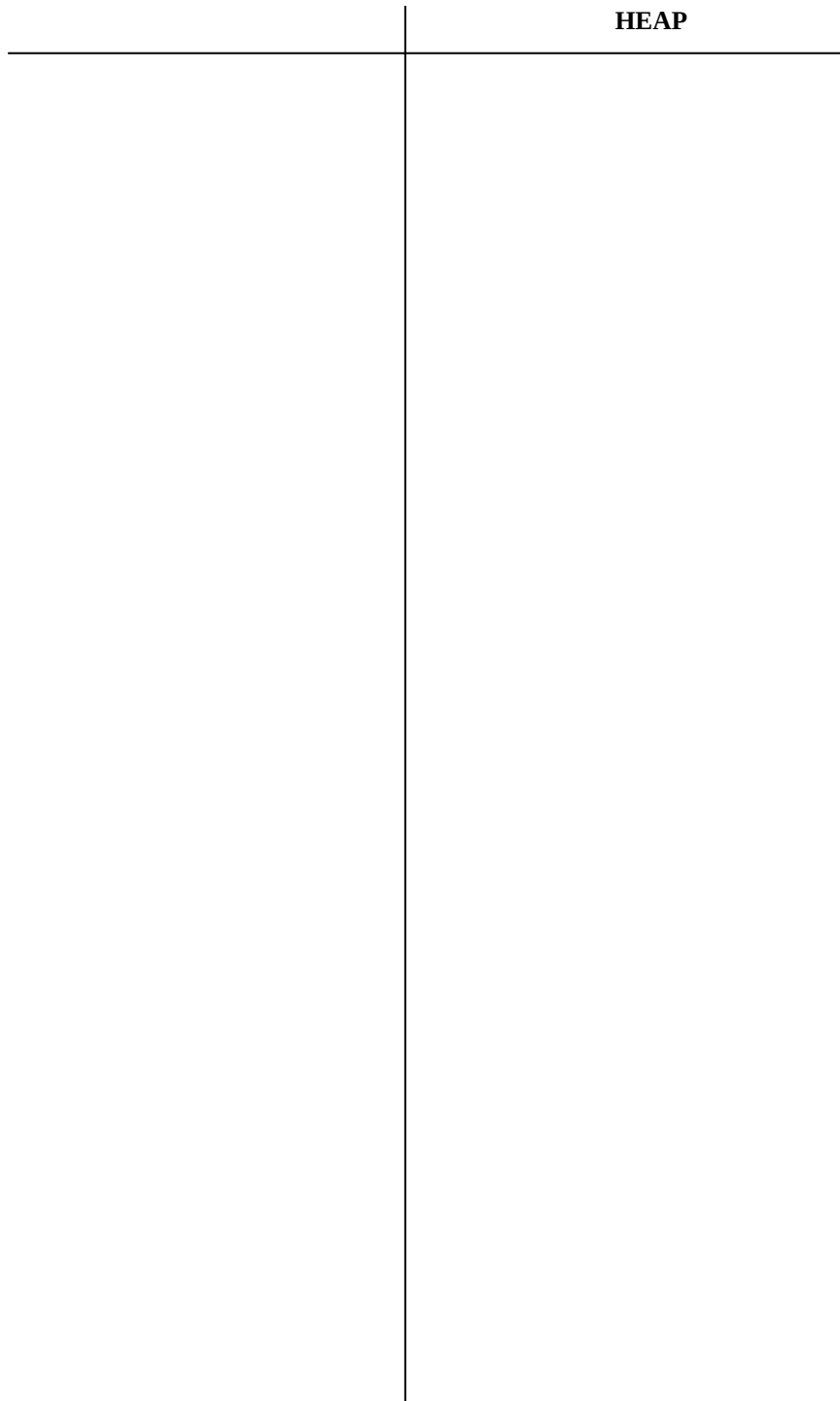
String messageOne = "Hello";
if (messageOne.equals("HELLO")) {
    System.out.println("Hello processed");
} else {
    System.out.println("Hello ignored");
}

String messageTwo = new String("Hello");
if (messageOne.equals(messageTwo)) {
    System.out.println("CaseOne");
} else {
    System.out.println("CaseTwo");
}

if (messageOne == messageTwo) {
    System.out.println("CaseThree");
} else {
    System.out.println("CaseFour");
}
```

b. (8 pts) Draw the memory map associated with the following code segment.

```
double pressure = 2.0;
String one = new String("Peter");
String two = new String("Rose");
String three = one;
String four = new String("Jackie");
two = null;
```



Problem 3 (35 pts)

- a. (10 pts) Write **PSEUDOCODE** for a program that reads integer values and computes the sum of those values. The program will stop reading values and will print the sum, after the user inputs the value 0. The message "Input Value" should be use to read the values, and the message "Sum is: " should precede the computed sum.
- b. (25 pts) Write a complete Java class named **Sum** that implements (in the main method) the **pseudocode** you design above. On the last page of the exam, you will find a description of methods you might find useful during the implementation of this class. You don't necessarily need to use all the methods provided on the last page.

WRITE YOUR ANSWERS ON THE NEXT PAGE.

AREA FOR ANSWERS TO QUESTION #3.

Problem 4 (34 points)

For this problem, you will implement a program which process images as long as there is money available to cover processing costs . This problem is based on the picture infrastructure from homework #2.

The program will first read a value representing the budget assigned for processing images. The program will then proceed to process images as long as the budget allows it. The cost of processing one image is \$1.00, which will be deducted from the remaining budget. The steps associated with processing one image are:

- a. Read the image name (location).
- b. Display the picture associated with the name read in a. We considered this picture the original picture.
- c. Posterize and double the original picture and display the resulting picture.

Assumptions/Restrictions

The following assumptions/restrictions apply to this problem:

- a. Implement the image processing system in the main method of a class named **PicturesManager**.
- b. You can assume the user will always provide a valid budget value (a String representing a valid double type value).
- c. You must use meaningful variable names and good indentation.
- d. Remember, the program is expected to process several images, as long as the budget allows it.
- e. Use the message "Enter Budget" when reading the budget value.
- f. Use the message "Enter Image" when reading the image name.

At the end of this description, and on the last page of the exam, you will find a description of methods and classes you might find useful during the implementation of this program. You don't necessarily need to use all the methods provided on the last page. By the way, you don't need to provide pseudocode for this problem

Overview of methods and class you might need

Image - This class allow us to convert a jpeg file into a picture object we can manipulate in our programs. When we create an Image object we need to provide a reference to a String object representing the image name (location).

PictureUtil – The show method of this class allow us to display a picture.

DoubleSize - This class enable us to double the size of a picture. When we create a DoubleSize object we need to provide a picture object reference as a parameter value.

Posterize - This class creates a picture object that "posterizes" an input picture. When we create a

Posterize object we need to provide a picture object reference as a parameter value.

Support Classes and Methods

JOptionPane.showInputDialog - It has as parameter the message to display on the dialog box.
It returns a reference to the String entered by the user.

JOptionPane.showMessageDialog - It has two parameters: the first one will be set to null, and
the second one to the String to display on the dialog box.

Double.parseDouble - It has as parameter a String reference. Returns a double initialized to
the value represented by the specified String.

Integer.parseInt - It has as parameter a String reference. Returns an integer value initialized to
the value represented by the specified String.