Computer Science 161 C

First Hour Exam

Name _____

Friday, Sept. 26, 2008
100 Pts.

I. Some Definitions

Give brief definitions of the following terms. It is not sufficient to simply say what the letters mean (though that will give you some partial credit.)

a. ALU

b. Control Unit

c. Accessor method

d. Instance variable

(definitions continued)

e. Constructor (method)

f. Memory (give a brief explanatory model)

II Some short answer questions (5 pts. each)

a. What is the relationship between classes and objects?

Write Java language expressions for the following (view '=" as an assignment statement)

b. $x = \frac{a+b}{c}$

c. $x = b^2 - 4ac$ (where a, b, and c are separate variables)

(short answer questions continued)

What will be the value of the variable x after the following calculations (answer with a number)

d. `x = 3*2-1;`

e. `x = 6/3*2`

f. What is the role of a variable in our model for memory?

III Java programming

A note on writing programs on an exam: Usually I ask for full documentation on the programs you write. However, during an exam you do not need to write comments except where I explicitly call for them.

a. (10 pts.) Consider a class named **Part** with instance variables (fields) **partName** (a **String**), **amtOnHand** (a **float**), **unitCost** (a **float**) and **reorderLevel** (also a **float** - remember that the data type float stores numbers with decimals). These are private fields. Write the appropriate Class statement to get this started (i.e., the class declaration and declaration of instance variables - everything except for the definition of methods, which follows after this question).

b. (15 pts.) All classes have at least one constructor. Write a constructor for the **Part** class which takes a part name, amount on hand, unit cost, and reorder levels as arguments and which initializes the instance variables.

d. (15 pts.) Fill in the Java code details of the following method. The method is to compare the amtOnHand against the reorderLevel. If amtOnHand is greater than or equal to reorderLevel, print a message (using System.out.println) that the part does not need to be reordered (be sure to include the name of the part in your message). If the amtOnHand is less than the reorderLevel, write a message that says that the stock on hand is low for this part (again naming the part) and say how much needs to be reordered so that the amtOnHand will be at twice the reorderLevel.

```
public void checkReorder()
{









}
```