Midterm 1

CS 1323, Fall 2015, Sections 10 and 995

Name (printed legibly):

Student number:

Section number (10 is at 9:30 a.m., 995 is online)

Integrity Pledge

On my honor, I affirm that I have neither given nor received inappropriate aid in the completion of this exercise.

Signed (type initials for student taking examination online):

Answer all programming questions in Java.

Unless otherwise indicated, each part of a problem is worth the same number of points.

Show your work to receive partial credit.

Pay careful attention to whether the question asks for a code fragment or a complete program. Do not write a whole program when you are asked for only a few lines of code.

Also pay attention to whether you need to get input from a user or not. When you do not need to get input, the problem will say something like “you may assume that the variables’ value was set somewhere else”.

Pay careful attention to distinctions like int versus double and String versus char.

Try to move through short problems quickly to leave you sufficient time to write programs and code fragments.

You will have fifty minutes to take the examination.

You do not need to use import statements on any code.

You may abbreviate System.out.println as S.o.p.

1. (10 points; 2 points each)

What type of data (int, double, String, char, or boolean) would you use to store each of the following things? Do not assume that each type is used exactly once.

* 1. Whether or not your car was recalled this week.
  2. The number of Volkswagen cars that were recalled this week.
  3. The name of the model of car that was recalled.
  4. The price per share of Volkswagen stock after this discovery, in dollars.
  5. The answer to the question "Would you buy a Volkswagen diesel car this week? Yes/No"

1. (10 points; 2 points each) Give the value computed for each expression below. Pay careful attention to type, especially char versus String and int versus double.
   1. 39 / 4
   2. 39 % 4
   3. 39 / (double) 4
   4. "123" + "456"
   5. 23 >= 92
2. (20 points; 4 points each part) Find the value assigned by each statement below. Show all intermediate steps to get partial credit. Each part is independent, with the values for any variables starting with the ones given below (do not use the results of a) in b), for example). If the expression is not legal in Java, say so.

Be sure to distinguish double and int values by **giving double values a decimal point**, even if it is a zero.

int width = 13;

double height = 9.5;

* 1. width = width \* 2 + 10;
  2. height = width / 3 \* 9 % 7;
  3. double circumference = 2 \* width + 2 \* height;
  4. width = width + height;
  5. width = (int) height + width;

1. (5 points) Your company sells advertisers a location on websites. Advertisers pay a flat fee of $229.99 for their advertisement, and an additional fee based on how many times the webpage was displayed to potential customers (called a per-click fee). The per-click fee is $1.99 per 1000 clicks. The full 1000 clicks has to be received to be paid. This means that if 999 clicks occur, there is no fee.

Write a ***code fragment*** that calculates how much money your company will make from a single client using the variables below.

int clicks; // This variable was given a value somewhere else

double cost; // this is amount of money your firm will make from this client

1. (10 points) Complete the ***code fragment*** started below that calculates the cost of a plane ticket. The ticket has a base cost, adds on $25 for the first suitcase and $35 for additional suitcases, and a $20 extra if you want to sit in a premium seat. Use the variables below.

double base; // the base cost of the ticket

int suitcases; // the number of suitcases

String premium; // Yes if the user wants a premium seat, no otherwise

double cost; // the cost of the ticket

Scanner input; // Assume this is previously constructed

System.out.println(“What is the base cost of your ticket?”); // input base below

System.out.println(“How many suitcases need to be checked?”); // input suitcases below

System.out.println(“Do you want to sit in a premium seat? Yes/No”); // input premium below

// Perform cost calculation below

1. (15 points) ***Trace the code fragments*** below in the tables at the right. If there is an infinite loop, trace three iterations and write “infinite loop” in the table.

a)

|  |
| --- |
| **size** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

int size = 8;

if (size > 10)

size = size + 3;

else

{

size = size – 2;

if (size > 4)

size = 2 \* size;

}

b)

int size = 9;

int sum = 0;

|  |  |
| --- | --- |
| **size** | **sum** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

while (size <27)

{

sum = sum + size;

size = size + 5;

}

c)

|  |  |
| --- | --- |
| **count** | **accumulator** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

int count = 20;

int accumulator = 0;

while (count < 15)

{

count = count + 5;

accumulator = accumulator – count;

}

1. (30 points) You work for the Pear company that produces the jPhone. Their phones are produced in batches of 1000. Each batch has to be drop tested to see if the phones can survive a drop from four feet onto cement. This kind of testing is called destructive testing because once the phone has been dropped, whether it is broken or not, it cannot be sold to customers. So Pear does sampling, which means that they only test 10 jPhones out of every 1000 produced. If 8 or more of the jPhones survive the drop test, then the batch is approved for shipping. If 7 or less jPhones survive, the batch is rejected.

Write a ***complete program*** that keeps track of the results of testing. Your program should report the percent of batches that were approved and rejected. The ***exact interaction*** that your program should produce is below. Numbers in italics came from the user.

Welcome to Pear's jPhone Testing System

How many phones passed testing in this batch? Enter 1000 to stop

*7*

How many phones passed testing in this batch? Enter 1000 to stop

*9*

How many phones passed testing in this batch? Enter 1000 to stop

*8*

How many phones passed testing in this batch? Enter 1000 to stop

*1000*

Percent of batches approved: 66.66666666666667%

Percent of batches rejected: 33.333333333333336%