Midterm 1 Solution

CS 1323, Fall 2015, Sections 10 and 995

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.

boolean

* 1. The number of Volkswagen cars that were recalled this week.

int

* 1. The name of the model of car that was recalled.

String

* 1. The price per share of Volkswagen stock after this discovery, in dollars.

double

* 1. The answer to the question "Would you buy a Volkswagen diesel car this week? Yes/No"

String

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

9

* 1. 39 % 4

3

* 1. 39 / (double) 4

39.0 / 4.0

9.75

* 1. "123" + "456"

“123456”

* 1. 23 >= 92

false

1. (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;

width = 13 \* 2 + 10

width = 26 + 10

width = 36

* 1. height = width / 3 \* 9 % 7;

height = 13 / 3 \* 9 % 7

height = 4 \* 9 % 7

height = 36 % 7

height = 1

height = 1.0

* 1. double circumference = 2 \* width + 2 \* height;

circumference = 2 \* 13 + 2 \* 9.5

circumference = 26 + 19.0

circumference = 26.0 + 19.0

circumference = 45.0

* 1. width = width + height;

width = 13 + 9.5

Illegal, can’t assign double to int

* 1. width = (int) height + width;

width = 9 + 13

width = 22

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

cost = 229.99 + (clicks/1000) \* 1.99

1. (10 points) Write a ***code fragment*** 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. The program interaction is shown below:

*What is the base cost of your ticket?*

***443.20***

*How many suitcases need to be checked?*

***1***

*Do you want to sit in a premium seat? Yes/No*

***Yes***

*Your ticket will cost $488.20*

The Scanner below has been declared and previously constructed.

Scanner stdin; // already constructed elsewhere

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

double base = stdin.getDouble();

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

int suitcases = stdin.getInt();

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

String premium = stdin.getString();

if (suitcases <= 1)

{

cost = base + suitcases \* 25;

}

else

{

cost = base + 25 + 35 \* (suitcases-1);

}

if ( premium.equals(“Yes”)) // could also be nested—but it needs to be in both parts

{

cost = cost + 20;

}

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** |
| 8 |
| 6 |
| 12 |
|  |
|  |
|  |
|  |

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** |
| 9 | 0 |
| 14 | 9 |
| 19 | 23 |
| 24 | 42 |
| 29 | 66 |
|  |  |
|  |  |

while (size <27)

{

sum = sum + size;

size = size + 5;

}

c)

|  |  |
| --- | --- |
| **count** | **accumulator** |
| 20 | 0 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

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%

**import** java.util.Scanner;

**public** **class** Pear

{

**public** **static** **void** main(String[] args)

{

Scanner keyboard = **new** Scanner(System.***in***);

System.***out***.println("Welcome to Pear's jPhone Testing"

+ " System");

**int** batchesTested = 0;

**int** batchesAccepted = 0;

**int** batchesRejected = 0;

**final** **int** THRESHOLD = 8;

System.***out***.println("How many phones passed testing "

+ "in this batch? Enter 1000 to stop");

**int** result = keyboard.nextInt();

**while** (result != 1000)

{

**if** (result >= THRESHOLD)

batchesAccepted = batchesAccepted + 1;

**else**

batchesRejected = batchesRejected + 1;

batchesTested = batchesTested + 1;

System.***out***.println("How many phones passed testing "

+ "in this batch? Enter 1000 to stop");

result = keyboard.nextInt();

}

System.***out***.println("Percent of batches approved: "

+ batchesAccepted\*100/(**double**) batchesTested

+ "%");

System.***out***.println("Percent of batches rejected: "

+ batchesRejected\*100/(**double**) batchesTested

+ "%");

}

}