Midterm 1 Solution

CS 1323-1324, Fall 2018

Name (printed legibly): _____

Student number: _____

Integrity Pledge

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

Signed: _____

Do not write on the back of pages.

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 elsewhere".

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 assume that the user enters all data perfectly unless otherwise noted.

You may abbreviate System.out.println as S.o.p., and may abbreviate prompts.

1) (10 points; 2 points each) Write a declaration (type and identifier) to use to store each of the following things. Do not assume that each type is used exactly once.

a) Whether or not the ball in a tennis match should be served from the court on the player's left.

boolean serveOnLeft;

b) The number of spectators who attended the U.S. Open.

int spectators;

c) The number of tons of soda, including partial tons, which were sold during the finals of the U.S. Open.

double tonsOfSoda;

d) The score of a tennis game (scores in tennis are love, 15, 30, 40, deuce, and advantage).

String score;

e) The sizes of t-shirts sold during the U.S. Open (sizes include XS, S, M, L, XL, XXL, XXXL).

String size;

2) (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.** If the expression is not legal, say so. Characters and Strings must be properly distinguished by using single and double quotes.

a) 54 / 11

4

b) 54 % 11

10

c) 3.4 – 1.4

2.0

d) 4/5 + 1/5

0 + 0

0

e) "4/5" + "1/5"

"4/51/5"

3) (24 points; 6 points each part) Find the value assigned to result 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. Distinguish double and int values.**

```
int touchdown = 28;
double yards = 32.4;
int safety = 2;
int fieldGoal = 8;
double passingYards = 5.2;
```

a) int result = touchdown + safety * fieldGoal;

= 28 + 2 * 8
= 28 + 16
= 44

b) double result = yards + touchdown / fieldGoal;

= 32.4 + 28 / 8
= 32.4 + 3
= 32.4 + 3.0
= 35.4

c) int result = (int) touchdown + safety + passingYards;

= (int) 28 + 2 + 5.2
= 28 + 2 + 5.2
= 30 + 5.2
= 30.0 + 5.2
= 35.2 Illegal because a double cannot be assigned to an int

d) double result = touchdown % fieldGoal + -yards * passingYards / safety;

= 28 % 8 + -32.4 *5.2 / 2
= 4 + -32.4 * 5.2/ 2
= 4 + -168.48 / 2
= 4 + -168.48 / 2.0
= 4 + -84.24
= -80.24

4) (10 points; 4 points for a), 6 points for b)) ***Trace the code fragments*** below in the tables at the right. **Remember to include the initial values in the table.**

a)

```
int points = 7;

int threshold = 15;

if (points != threshold)
{
        points = points + 5;
}
else
{
        threshold = threshold – 5;
}
if (points < threshold)
{
        threshold = points – 2;
}
```

points	threshold
7	15
12	10

b)

```
double price = 35.27;
int discount = 0;
if (price > 50.0)  // Read this if/else statement very carefully
{
        discount = 1;
}
else if (price > 100.0)
{
        discount = 10;
}
else if (price > 500.0)
{
        discount = 50;
}
else
{
        discount = 100;
}
```

price	discount
35.27	0
	100

5) (16 points; 6 points for a), 3 points for b), 7 points for c)) Write code fragments, using the variables declared below, that solves the problem.

Sometimes it is cheaper to purchase things in larger quantities, but not always. For example, peanut butter comes in three size jars, described in the table below (right). If you needed 34 ounces of peanut butter, it would be cheaper to buy 2 small jars ($5.00) than to buy one medium jar ($5.78). Of course this assumes that you only need 34 ounces and have no use for extra peanut butter. This makes sense to me because I HATE PEANUT BUTTER.

Jar Size	Ounces	Price
Small	16	$2.50
Medium	50	$5.78
Large	64	$8.87

```
int ouncesNeeded; // value given elsewhere
int smallJars; // the number of small jars you need to purchase
double smallCost; // The cost of purchasing only small jars
double mediumCost; // The cost of purchasing only medium jars
double largeCost; // The cost of purchasing only large jars
double minimumCost; // Set this variable to the price of the least costly option
```

a) Using data and variables from above, find the number of jars necessary for the small size.

```
smallJars = (int) Math.ceil(ouncesNeeded/16.0);
```

b) Find the cost of the smallJars by declaring and using an appropriate constant.

```
double final SMALL_COST = 2.50;
smallCost = smallJars * SMALL_COST;
```

c) If the variables smallCost, mediumCost and largeCost have all been set, show code that calculates minimumCost

```
minimumCost = smallCost;
if (mediumCost < minimumCost)
{
        minimumCost = mediumCost;
}
if (largeCost < minimumCost)
{
        minimumCost = largeCost;
}
```

-or-

```
minimumCost = Math.min(smallCost, mediumCost);
minimumCost = Math.min(minimumCost, largeCost);
```

6) (30 points) Write a complete program that helps a user interpret data collected by a FitByte wearable exercise tracker.[1]

You must use at least one meaningful constant in this program.

FitBytes count steps by using sensors to analyze movement. Once the steps are counted, they need to translate the steps to a distance traveled. The table below summarizes the length of a single step as a function of age.

30 or younger	**31-50**	**51-70**	**71 or older**
30”	28”	24”	23”

While strides are given in inches, people want data to be displayed in miles. One mile equals 63360 inches. So to convert inches to miles, you divide by 63,360.

Two example runs of the program are shown below (user input in bold italics):

```
Enter the number of steps walked
10000
Enter your age
27
You have walked 4.734848484848484 miles

Enter the number of steps walked
2000
Enter your age
47
You have walked 0.8838383838383839 miles
```

Do not write anything on this page—start you code on the next page.

[1] The information in this project was based on https://help.fitbit.com/articles/en_US/Help_article/1141.

```java
import java.util.Scanner;
public class FitByte {

	public static void main(String[] args)
	{
		Scanner keyboard = new Scanner(System.in);

		final int INCHES_IN_MILE = 63360;

		System.out.println("Enter the number of steps walked");
		int steps = keyboard.nextInt();
		keyboard.nextLine();

		System.out.println("Enter your age");
		int age = keyboard.nextInt();
		keyboard.nextLine();

		int strideInches;
		if (age <= 30)
		{
			strideInches = 30;
		}
		else if (age <= 50)
		{
			strideInches = 28;
		}
		else if (age <= 70)
		{
			strideInches = 24;
		}
		else
		{
			strideInches = 23;
		}

		int distanceInches = steps * strideInches;

		double distanceMiles = distanceInches / INCHES_IN_MILE;

		System.out.println("You have walked " + distanceMiles
			+ " miles");
		keyboard.close();
	}

}
```