

Midterm 1

Introduction to Programming

Sept. 16, 2019

Name (printed):

Solution

Student ID:

Integrity Pledge

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

Signed: _____

Instructions: Answer all programming questions in Java. For a given problem, each part is worth the same number of points unless stated otherwise. Show your work to receive partial credit.

Additional notes:

- You have fifty minutes to complete the exam.
- Pay attention to distinctions like int versus double and String versus char.
- Try to complete short problems quickly so you have enough time to write programs and code fragments.
- Pay attention to whether each programming question is asking for a code fragment or a complete program. Do not write a whole program when you are only asked for a few lines of code.
- Pay attention to whether you need input from a user. When you do not need to get input, the problem will say something like “you may assume that the value of the variable was set elsewhere.”
- You do not need to include import statements in any code. You may assume that the user enters all data perfectly unless stated otherwise.
- You may abbreviate `System.out.println` as `S.o.p.`, and you may abbreviate prompts.

Do not write on the back of any page.

Problem 1 (12 points; 2 per part): Declare variables to store the information described below in a Java program. For each part, declare a single variable with a well-chosen type and a meaningful, legal identifier. Do not assign any data to the variables.

- a) Whether or not Starbucks is currently selling Pumpkin Spice Lattes.

```
boolean isSellingPSL;
```

- b) The size of a Pumpkin Spice Latte abbreviated as either T (tall), G (grande), or V (venti).

```
char sizePSL; // String will also work
```

- c) The price of a grande Pumpkin Spice Latte in dollars.

```
double pricePSLDollars;
```

- d) The month and day when Pumpkin Spice Lattes went on sale this year. (The month will not be abbreviated, and it will be separated from the day by a space.)

```
String datePSLOnSale;
```

- e) The number of Pumpkin Spice Lattes sold last year.

```
int numPSLSold2018;
```

- f) The number of whole grams of sugar in a tall Pumpkin Spice Latte.

```
int gramsSugarTallPSL;
```

Problem 2 (14 points; 2 per part): Give the result computed for each expression below. Pay attention to the order of operations and the data type of the result. Characters and Strings must be properly distinguished by using single and double quotes respectively. If the expression is not legal, say so.

a) $42 / 9$

4

b) $48 \% 7$

6

c) $12.5 * 2$

25.0

d) `"2019" + "09"`

`"201909"`

e) $17 / 4 + 3 \% 8$

$4 + 3 \% 8$

$4 + 3$

7

f) `(int) 19.99`

19

g) `(double) 4 / 5 - 5`

$4.0 / 5 - 5$

$4.0 / 5.0 - 5$

$0.8 - 5$

$0.8 - 5.0$

-4.2

Problem 3 (20 points; 4 per part): Find the value assigned to the variable result by each statement below. Show all intermediate steps to get partial credit. If the expression is not legal in Java, write "Illegal." Make sure you distinguish double and int values.

```
int bamboo = 7;  
int succulents = 4;  
double moneyTree = 35.2;  
int iris = 10;  
int bonsai = 9;
```

a) `int result = bamboo + iris / succulents;`

```
7 + 10 / 4  
7 + 2  
9
```

b) `int result = bonsai + succulents * succulents % bamboo - iris;`

```
9 + 4 * 4 % 7 - 10  
9 + 16 % 7 - 10  
9 + 2 - 10  
11 - 10  
1
```

c) `double result = bamboo * bonsai + moneyTree;`

```
7 * 9 + 35.2  
63 + 35.2  
63.0 + 35.2  
98.2
```

d) `int result = Math.ceil(moneyTree);`
`// Documentation for this method is given below.`

```
Math.ceil(35.2)
36.0 // Math.ceil returns a double
```

Illegal: a double value cannot be assigned to an integer variable, even if the fractional part of the value is 0.

e) `double result = (bamboo * bamboo % iris - bonsai) * moneyTree;`

```
(7 * 7 % 10 - 9) * 35.2
(49 % 10 - 9) * 35.2
(9 - 9) * 35.2
0 * 35.2
0.0 * 35.2
0.0
```

ceil

```
public static double ceil(double a)
```

Returns the smallest (closest to negative infinity) double value that is greater than or equal to the argument and is equal to a mathematical integer. Special cases:

- If the argument value is already equal to a mathematical integer, then the result is the same as the argument.
- If the argument is NaN or an infinity or positive zero or negative zero, then the result is the same as the argument.
- If the argument value is less than zero but greater than -1.0, then the result is negative zero.

Note that the value of `Math.ceil(x)` is exactly the value of `-Math.floor(-x)`.

Parameters:

a - a value.

Returns:

the smallest (closest to negative infinity) floating-point value that is greater than or equal to the argument and is equal to a mathematical integer.

Problem 4 (12 points; 6 per part): Trace the execution of the following statements by filling in the table to the right of each code fragment. Each column tracks the values stored in a different variable. Note that the tables may contain more rows than you need. **Be sure to store the initial assignments in the tables.**

```
a) int length = 15;
   if (length > 5 && length < 30)
   {
       length = length * 2 + 1;
   }
   else
   {
       length = length + 2;
   }
   if (length >= 30)
   {
       length = length + 7;
   }
```

length
15
31
38

```
b) double purchase = 125.95;
   int cashRewards = 0;
   char customerLevel = '-';
   if (purchase >= 75)
   {
       cashRewards = 10;
       customerLevel = 'C';
   }
   else if (purchase >= 95) {
       cashRewards = 20;
       customerLevel = 'B';
   }
   else if (purchase >= 125) {
       cashRewards = 30;
       customerLevel = 'A';
   }
   else
   {
       cashRewards = 50;
       customerLevel = 'S';
   }
```

purchase	cashRewards	customerLevel
125.95	0	'-'
	10	'C'

Problem 5 (12 points; 6 per part): Write code fragments using the given variables to calculate and store the quantities described below.

I'm ordering hamburgers and lemonade for a party! Help me calculate the total cost of the hamburgers and the amount of lemonade I can afford.

- a) Calculate the total cost of the hamburgers. Use the variables storing the price per burger, the number of guests coming to the party, the number of burgers ordered for each guest, and the number of extra burgers ordered in case additional people attend.

```
double totalCostOfBurgers; // Store the result in this variable.

// The following variables are given a value elsewhere in the program.
double pricePerBurger; // Price of each burger.
int guests;           // Number of guests coming to the party.
int burgersPerGuest;  // Number of burgers ordered for each guest.
int extraBurgers;     // Number of extra burgers ordered.
```

```
totalCostOfBurgers =
    (burgersPerGuest * guests + extraBurgers) * pricePerBurger;
```

- b) Calculate the number of liters of lemonade I can afford to purchase given the amount of money I've already spent on hamburgers. Use the variables storing my party budget, the price per liter of lemonade, and the total cost of the hamburgers, which you assigned in part (a). Round the number of liters down to the nearest liter.

```
int litersOfLemonade; // Store the result in this variable.

// The following variables are given a value elsewhere in the program.
double partyBudget;   // Total amount of money for the party.
double pricePerLiter; // Price of each liter of lemonade.
```

```
litersOfLemonade =
    (int) ((partyBudget - totalCostOfBurgers) / pricePerLiter);
```

Problem 6 (30 points): Write a complete program that collects three grades and a name from the user and outputs the average grade and the corresponding letter grade. Assume the grades are input as integers between 0 and 100.

The following table shows the range of average grades that corresponds to each letter grade:

Average Grade	Letter Grade
$90 \leq \text{avg} \leq 100$	A
$80 \leq \text{avg} < 90$	B
$70 \leq \text{avg} < 80$	C
$40 \leq \text{avg} < 70$	D
$0 \leq \text{avg} < 40$	F

For example, an average grade that is greater than or equal to 80 and less than 90 is a B.

Below is an example run of the program with user input underlined and in bold. Note that student names can contain spaces. The output of your code should match this output if given the same input.

```
Enter grade 1
75
Enter grade 2
92
Enter grade 3
86
Enter student name
James Dizikes
Grade summary for James Dizikes
Average: 84.33333333333333
Letter: B
```

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


```
import java.util.Scanner;

public class AverageGrade {

    public static void main(String[] args) {

        // Construct a Scanner to read input from the keyboard.
        Scanner keyboard = new Scanner(System.in);

        // Get the three grades from the user.
        System.out.println("Enter grade 1");
        int grade1 = keyboard.nextInt();

        System.out.println("Enter grade 2");
        int grade2 = keyboard.nextInt();

        System.out.println("Enter grade 3");
        int grade3 = keyboard.nextInt();

        // Skip the newline character after the third grade before reading
        // the name of the student using nextLine.
        keyboard.nextLine();

        System.out.println("Enter student name");
        String name = keyboard.nextLine();

        // Calculate the average grade.
        double averageGrade = (grade1 + grade2 + grade3) / 3.0;

        // Determine the corresponding letter grade. Note that the variable
        // letterGrade must be declared outside of the conditional statements
        // so that it can be used later in the print statement.
        char letterGrade;

        if (averageGrade >= 90) {
            letterGrade = 'A';
        }
        else if (averageGrade >= 80) {
            letterGrade = 'B';
        }
    }
}
```

```
        else if (averageGrade >= 70) {
            letterGrade = 'C';
        }
        else if (averageGrade >= 40) {
            letterGrade = 'D';
        }
        else {
            letterGrade = 'F';
        }

        // Output information about the grades to the user.
        System.out.println("Grade summary for " + name);
        System.out.println("Average: " + averageGrade);
        System.out.println("Letter: " + letterGrade);

    } // End of main method
} // End of class
```