

Assignment #2

CSE110 - Arizona State University

Topics

- To declare and use variables
- Arithmetic Expressions
- Using the Scanner class to get the input from the user
- Use selection statements to find a maximum value

Coding Guidelines:

- Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
- Keep identifiers to a reasonably short length.
- User upper case for constants. Use title case (first letter is upper case) for classes. Use lower case with uppercase word separators for all other identifiers (variables, methods, objects).
- Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
- Use white space to make your program more readable.

Important Note

All submitted assignments must begin with the descriptive comment block. To avoid losing trivial points, make sure this comment header is included in every assignment you submit, and that it is updated accordingly from assignment to assignment. Reasonably good amount of comments should be added in your program so that it is easy for other people to understand it. Please see the comment style in the textbook.

Part #1: Written Exercises (5 pts)

1. (2 pts.) What do the following expressions evaluate to in Java given `int x = 3, y = 6;`
 - a) `x == y / 2`
 - b) `x % 2 == 0 || y % 2 != 0`
 - c) `x - y < 0 && !(x >= y)`
 - d) `x + 6 != y || x / y <= 0`
2. (1 pt.) Write statements to prompt for and read the user's full name using a `Scanner` variable.
3. (2 pts.) What does the following statement sequence print? (page 63)

```
String str = "Harry";
int n = str.length();
String mystery = str.substring(0,1) + str.substring(n-1, n);
System.out.println(mystery);
```

Part #2 - Programming (15 pts)

Overview

Write a complete Java program in a source file to be named `Assignment2.java`. The program prints out the following menu to a user, using `'\t'` (tabs) and `'\n'` (newline), and `print` or `println` methods of `System.out`:

Welcome to the In-N-Out Burger menu:

```
-----  
Hamburger $2.75  
Cheeseburger $3.25  
French Fries $2.50  
Shake & Beverage: $1.50
```

Then prompts the user:

How many hamburger(s) would you like to buy?

and read in the number. Then prompts the user:

How many cheeseburger(s) would you like to buy?

and read in the number. Then prompts the user:

How many French fries would you like to buy?

and read in the number. Then prompts the user:

How many drinks would you like to buy?

and read in the number. Then the program computes and prints the following:

- The total cost for the hamburger(s)
- The total cost for the cheeseburger(s)
- The total cost for fries
- The total cost for drink(s)
- Which item had the highest total cost
- The total cost for the order
- The total number of hamburgers, cheeseburgers, French fries, and drinks ordered

Hint

One way to find which item has the highest total cost is to create a `String` variable to hold the name of that item and then use comparisons to check if the cost of hamburgers is greater than the other prices (using `&&`) then set the `String` to "hamburgers". You would need a similar check to see if the other total costs are the largest. Then you could output the highest `String` when needed (see sample output below).

To earn full points on this assignment you will have to:

- Use constant doubles for price of a Hamburgers, Cheeseburger, French Fries and Drinks.

- For example, you would have to use a constant for the price of hamburgers similar to

```
* final double HAMBURGER_PRICE = 2.75;
```
- and then use that constant where needed in the program. Be sure to follow naming conventions (ALL.CAPS) when naming constants.
- Use the `NumberFormat` class to format dollar amounts to be printed.
 - The `NumberFormat` class is defined in (and needs to be imported from) the `java.text.NumberFormat` package.
 - See the example video of `Purchased.java` posted on the course website under “Week 1” → “Lectures: Chapter 2” → “NumberFormat Example”).

Output Sample 1

(user input is in bold)

Welcome to the In-N-Out Menu:

Hamburger: \$2.75
 Cheeseburger: \$3.25
 French Fries: \$2.50
 Shakes & Beverages: \$1.50

How many hamburger(s) would you like to buy? **3**
 How many cheeseburger(s) would you like to buy? **4**
 How many French fries would you like to buy? **3**
 How many drink(s) would you like to buy? **3**

Total cost for the hamburger(s): \$8.25
 Total cost for the cheeseburger(s): \$13.00
 Total cost for the fries: \$7.50
 Total cost for the drink(s): \$4.50
 The cheeseburgers had the highest total cost.

Total cost for your order: \$33.25
 Total number of item(s) ordered: 13

Output Sample 2

(user input is in bold)

Welcome to the In-N-Out Menu:

Hamburger: \$2.75
 Cheeseburger: \$3.25
 French Fries: \$2.50
 Shakes & Beverages: \$1.50

How many hamburger(s) would you like to buy? **2**
 How many cheeseburger(s) would you like to buy? **1**
 How many French fries would you like to buy? **3**
 How many drink(s) would you like to buy? **3**

Total cost for the hamburger(s): \$5.50

Total cost for the cheeseburger(s): \$3.25
Total cost for the fries: \$7.50
Total cost for the drink(s): \$4.50
The fries had the highest total cost.

Total cost for your order: \$20.75
Total number of item(s) ordered: 9

Submission

- Go to the course web site (my.asu.edu), and then click on the on-line Submission tab.
- Submit your Assignment2.java file on-line. Make sure to choose Hw2 from drop-down box.
- Assignment2.java should have the following, in order:
 - In comments, the assignment header.
 - In comments, the answers to questions in Part #1.
 - The working Java code requested in Part #2.
 - The Assignment2.java file must compile and run as you submit it. You can confirm this by viewing your submission results.

Important Note: You may resubmit as many times as you like until the deadline, but we will only mark your last submission.

NO LATE ASSIGNMENTS WILL BE ACCEPTED.