



Drink Order Program

by Sophia



WHAT'S COVERED

In this lesson, you will learn about how to use conditional statements and their outcomes in Java.

Specifically, this lesson covers:

[1. Planning the Algorithm](#)

[2. Writing the Code](#)

1. Planning the Algorithm

The program that you will build in this tutorial is designed to take in drink orders from users. Users will be able to choose different types of options based on the drink that they select.

Take a look at the basic menu given to us by the company that has requested the following choices, based on what they offer in their restaurant:

Drink Menu

Water

Hot

Cold

Ice/No Ice

Coffee

Decaffeinated or Not?

Milk or Cream or None

Sugar or None

Tea

Green

Black

Considering these options, the user should be first prompted with the choice of water, coffee and tea. Based on each of those selections, there are additional prompts that are unique to each beverage. As the user enters in each option, a string should be built to indicate the entire order to be output at the end.



THINK ABOUT IT

Before you get into any coding, look at the selections again. What questions do you envision that we need to ask the user? Using the user's input, what conditionals do you think will be needed? What outputs would you expect the user to see? Given the input, what should happen next?

These questions help formulate the algorithm for this program.



BRAINSTORM

Directions: Pseudocode is an English-like statement that describes the steps in a program or algorithm. Write down the steps in pseudocode that you believe will be needed to make this program successful.



HINT

For the final project in this course you will identify a problem and create a program to solve it. Now is a great time to start “flexing” your algorithm step-building muscles.

2. Writing the Code

Given the structure of the drink selection items, there are a series of selection statements that will need to be used to build the output string.



TRY IT

Directions: The first step is to think about the initial check of the types of drinks. Enter the following code that initially checks the drink types. As you enter the code (in a file named DrinkOrder.java), add any comments that you think will help you better understand what is happening. Take note of the fact that we are using the escape character with the new line character “\n”. Remember the escape character in the previous Challenge? This feature allows you to write a single line that will present as multiple lines in the output since the \n adds a new line after each use of \n.

IN CONTEXT

Programming the Initial Check for Drink Types

```
import java.util.Scanner;

public class DrinkOrder {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
```

```

System.out.println("What type of drink would you like to order?");
// Note use of new line \n to print 3 lines with 1 statement.
System.out.println("1. Water\n2. Coffee\n3. Tea");
System.out.print("Drink selection #: ");
// String variable to hold drink details
String drinkDetails = "No drink chosen.";
int choice = input.nextInt();
// Remove \n left in input to avoid problems with later inputs
input.nextLine();
if(choice == 1) {
    drinkDetails = "Water";
}
else if(choice == 2) {
    drinkDetails = "Coffee";
}
else if(choice == 3) {
    drinkDetails = "Tea";
}
else {
    System.out.println("Sorry, not a valid drink selection.");
}
// Print out final drink selection
System.out.println("Your drink selection: " + drinkDetails + ".");
}
}

```

The DrinkOrder.java program should look like this:

What type of drink would you like to order?

1. Water
2. Coffee
3. Tea

Drink selection #: 1

Your drink selection: Water.



REFLECT

This provides the initial parts to set the drink choices. We will build onto our basic program with practice below.



TRY IT

Directions: Now within each selection, break down those options. Start with the water selection. Prompt the user to determine if they want to have hot or cold water.



STEP BY STEP

Step 1: Enter the **bolded code** inside the water selection section of the sample below. Note that the `Water` conditional is assigning the variable `drinkDetails` as `Water`.

Step 2: Then, after the input of Hot or Cold, concatenate that response on the output string using the concatenation operator `+`. Remember, concatenation is the operation of joining or merging two or more strings together.

Step 3: Use the `+=` operator to add two values together and assign that final value to the variable.

IN CONTEXT

Adding the Check for Water Sub-Option Temperature

```
import java.util.Scanner;

public class DrinkOrder {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("What type of drink would you like to order?");
        // Note use of new line \n to print 3 lines with 1 statement.
        System.out.println("1. Water\n2. Coffee\n3. Tea");
        System.out.print("Drink selection #: ");
        // String variable to hold drink details
        String drinkDetails = "No drink chosen.";
        int choice = input.nextInt();
        // Remove \n left in input to avoid problems with later inputs
        input.nextLine();
        if(choice == 1) {
            drinkDetails = "Water";
            System.out.println("Would you like that 1) hot or 2) cold?");
            System.out.print("Enter temperature selection #: ");
            choice = input.nextInt();
            // Remove \n left in input to avoid problems with later inputs
            input.nextLine();
            if(choice == 1) {
                drinkDetails += ", hot";
            }
            else if(choice == 2) {
                drinkDetails += ", cold";
            }
            else {
                System.out.println("Not a valid temperature selection.");
            }
        }
    }
}
```

```

    }

    }
    else if(choice == 2) {
        drinkDetails = "Coffee";
    }
    else if(choice == 3) {
        drinkDetails = "Tea";
    }
    else {
        System.out.println("Sorry, not a valid drink selection.");
    }
    // Print out final drink selection
    System.out.println("Your drink selection: " + drinkDetails + ".");
}
}

```



Directions: You can test both options of hot and cold water as well as an incorrect option. Run the program testing each option.

⇒ **EXAMPLE** Testing user selection of Hot Water (Temperature Selection 1):

What type of drink would you like to order?

1. Water
2. Coffee
3. Tea

Drink selection #: 1

Would you like that 1) hot or 2) cold?

Enter temperature selection #: 1

Your drink selection: Water, hot.

⇒ **EXAMPLE** Testing user selection of Cold Water (Temperature Selection 2):

What type of drink would you like to order?

1. Water
2. Coffee
3. Tea

Drink selection #: 1

Would you like that 1) hot or 2) cold?

Enter temperature selection #: 2

Your drink selection: Water, cold.

🔗 EXAMPLE Testing user selection of "0" for temperature (Invalid Selection):

What type of drink would you like to order?

1. Water
2. Coffee
3. Tea

Drink selection #: 1

Would you like that 1) hot or 2) cold?

Enter temperature selection #: 0

Not a valid temperature selection.

Your drink selection: Water.



REFLECT

Did your program function as intended during testing? What happened when the user entered "0"? Since we didn't add a "0" option, the program returned an error message: "Not a valid temperature selection".

The next step to determine after the user selected cold water, is to find out if they would like ice.



TRY IT

Directions: Add the code to determine if the user wants ice added to their cold water (shown in bold text below). Now concatenate that decision onto the `drinkDetails` variable by modifying the line code to `drinkDetails += ", with ice";`.

IN CONTEXT

Adding the Check for Sub-Option Ice When User Selects Water Sub-Option Cold

```
import java.util.Scanner;

public class DrinkOrder {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("What type of drink would you like to order?");
        // Note use of new line \n to print 3 lines with 1 statement.
        System.out.println("1. Water\n2. Coffee\n3. Tea");
        System.out.print("Drink selection #: ");
        // String variable to hold drink details
        String drinkDetails = "No drink chosen.";
        int choice = input.nextInt();
        // Remove \n left in input to avoid problems with later inputs
        input.nextLine();
        if(choice == 1) {
            drinkDetails = "Water";
            System.out.println("Would you like that 1) hot or 2) cold?");
```

```

System.out.print("Enter temperature selection #: ");
choice = input.nextInt();
// Remove \n left in input to avoid problems with later inputs
input.nextLine();
if(choice == 1) {
    drinkDetails += ", hot";
}
else if(choice == 2) {
    drinkDetails += ", cold";
    System.out.print("Would you like ice? (Y/N) ");
    // Read input as a String
    String response = input.nextLine();
    // Extract 1st char
    char yesNo = response.charAt(0);
    // Y or y is yes, anything else interpreted as no
    if(yesNo == 'Y' || yesNo == 'y') {
        drinkDetails += ", with ice";
    }
}
else {
    System.out.println("Not a valid temperature selection.");
}

}
else if(choice == 2) {
    drinkDetails = "Coffee";
}
else if(choice == 3) {
    drinkDetails = "Tea";
}
else {
    System.out.println("Sorry, not a valid drink selection.");
}
// Print out final drink selection
System.out.println("Your drink selection: " + drinkDetails + ".");
}
}

```

HINT

Be sure to run the program again, testing each water option to ensure it still functions properly. Are the results as you intended?

The output should look like this:

```
What type of drink would you like to order?
1. Water
2. Coffee
3. Tea
Drink selection #: 1
Would you like that 1) hot or 2) cold?
Enter temperature selection #: 2
Would you like ice? (Y/N) y
Your drink selection: Water, cold, with ice.
```



REFLECT

Notice that we don't have to worry about the check if the user's selection is *No ice*, as we don't have to include that as part of the items to add. If we did want to include that, we can use the `else if()` to check if the user response is a `No` or not.



TRY IT

Directions: Since the tea selection is the easiest next option, we can fill that part out for the selection of the green or black tea. Now you will add the **bolded code** below to determine if the user wants green or black tea with a Tea drink choice decision.

IN CONTEXT

Adding the Check for Tea Selection Sub-Options

```
import java.util.Scanner;

public class DrinkOrder {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("What type of drink would you like to order?");
        // Note use of new line \n to print 3 lines with 1 statement.
        System.out.println("1. Water\n2. Coffee\n3. Tea");
        System.out.print("Drink selection #: ");
        // String variable to hold drink details
        String drinkDetails = "No drink chosen.";
        int choice = input.nextInt();
        // Remove \n left in input to avoid problems with later inputs
        input.nextLine();
        if(choice == 1) {
            drinkDetails = "Water";
            System.out.println("Would you like that 1) hot or 2) cold?");
```



```

System.out.print("Enter temperature selection #: ");
choice = input.nextInt();
// Remove \n left in input to avoid problems with later inputs
input.nextLine();
if(choice == 1) {
    drinkDetails += ", hot";
}
else if(choice == 2) {
    drinkDetails += ", cold";
    System.out.print("Would you like ice? (Y/N) ");
    // Read input as a String
    String response = input.nextLine();
    // Extract 1st char
    char yesNo = response.charAt(0);
    // Y or y is yes, anything else interpreted as no
    if(yesNo == 'Y' || yesNo == 'y') {
        drinkDetails += ", with ice";
    }
}
else {
    System.out.println("Not a valid temperature selection.");
}

}
else if(choice == 2) {
    drinkDetails = "Coffee";
}
else if(choice == 3) {
    drinkDetails = "Tea";
    System.out.print("Type of tea: 1) Black or 2) Green: ");
    int teaChoice = input.nextInt();
    // Remove \n left in input to avoid problems with later inputs
    input.nextLine();
    if(teaChoice == 1) {
        drinkDetails += ", black";
    }
    else if(teaChoice == 2) {
        drinkDetails += ", green";
    }
    else {
        // Invalid selection - assume black tea
        drinkDetails += ", black";
    }
}

```

```
        System.out.println("Not a valid tea choice. Assuming black tea.");
    }
}
else {
    System.out.println("Sorry, not a valid drink selection.");
}
// Print out final drink selection
System.out.println("Your drink selection: " + drinkDetails + "");
}
}
```



TRY IT

Directions: Run the program again, testing the Tea drink options.

⇒ **EXAMPLE** Testing user selection of black tea:

What type of drink would you like to order?

1. Water
2. Coffee
3. Tea

Drink selection #: 3

Type of tea: 1) Black or 2) Green: 1

Your drink selection: Tea, black.

⇒ **EXAMPLE** Testing user selection of green tea:

What type of drink would you like to order?

1. Water
2. Coffee
3. Tea

Drink selection #: 3

Type of tea: 1) Black or 2) Green: 2

Your drink selection: Tea, green.

Be sure to test both tea choices on your own.



REFLECT

That was the only criteria that was set up for the tea. Did you notice the additional lines of code to handle invalid selections? We decided to make the assumption that when a user makes an invalid tea type selection, they will receive black tea.

With the coffee option, there are three separate parts. However, they are very similar to one another.



Directions: Finally, add the code for the sub-options for coffee. The restaurant menu indicated that the options for coffee include decaf, milk, cream, and sugar. You can break these options up into a choice for decaf, a choice for milk or cream, and a choice for sugar. The **bolded text** below shows how to handle these choices in the drink program.

IN CONTEXT

Adding the Check for Coffee Selection Sub-Options

```
import java.util.Scanner;

public class DrinkOrder {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);    System.out.println("What type of drink would you like to
order?");
        // Note use of new line \n to print 3 lines with 1 statement.
        System.out.println("1. Water\n2. Coffee\n3. Tea");
        System.out.print("Drink selection #: ");
        // String variable to hold drink details
        String drinkDetails = "No drink chosen.";
        int choice = input.nextInt();
        // Remove \n left in input to avoid problems with later inputs
        input.nextLine();
        if(choice == 1) {
            drinkDetails = "Water";
            System.out.println("Would you like that 1) hot or 2) cold?");
            System.out.print("Enter temperature selection #: ");
            choice = input.nextInt();
            // Remove new line left in input stream to avoid problems with later inputs
            input.nextLine();
            if(choice == 1) {
                drinkDetails += ", hot";
            }
            else if(choice == 2) {
                drinkDetails += ", cold";
                System.out.print("Would you like ice? (Y/N) ");
                // Read input as a String
                String response = input.nextLine();
                // Extract 1st char
                char yesNo = response.charAt(0);
                // Y or y is yes, anything else interpreted as no
                if(yesNo == 'Y' || yesNo == 'y') {
```

```

        drinkDetails += ", with ice";
    }
}
else {
    System.out.println("Not a valid temperature selection.");
}
}
else if(choice == 2) {
    drinkDetails = "Coffee";
    System.out.print("Would you like decaf? (Y/N): ");
    String decafResponse = input.nextLine();
    char decafYesNo = decafResponse.charAt(0);
    if(decafYesNo == 'Y' || decafYesNo == 'y') {
        drinkDetails += ", decaf";
    }
    System.out.println("Would you like 1) milk, 2) cream, or 3) none?");
    System.out.print("Enter choice #: ");
    int milkCreamChoice = input.nextInt();
    // Remove new line left in input stream to avoid problems with later inputs
    input.nextLine();
    if(milkCreamChoice == 1) {
        drinkDetails += ", milk";
    }
    else if(milkCreamChoice == 2) {
        drinkDetails += ", cream";
    }
    System.out.print("Would you like sugar? (Y/N): ");
    String sugarResponse = input.nextLine();
    char sugar = sugarResponse.charAt(0);
    if(sugar == 'Y' || sugar == 'y') {
        drinkDetails += ", sugar";
    }
}
else if(choice == 3) {
    drinkDetails = "Tea";
    System.out.print("Type of tea: 1) Black or 2) Green: ");
    int teaChoice = input.nextInt();
    // Remove \n left in input to avoid problems with later inputs
    input.nextLine();
    if(teaChoice == 1) {
        drinkDetails += ", black";
    }
}

```

```

else if(teaChoice == 2) {
    drinkDetails += ", green";
}
else {
    // Invalid selection - assume black tea
    drinkDetails += ", black";
    System.out.println("Not a valid choice. Assuming black tea.");
}
}
else {
    System.out.println("Sorry, not a valid drink selection.");
}

// Print out final drink selection
System.out.println("Your drink selection: " + drinkDetails + ".");
}
}

```



REFLECT

For now you'll just test the one case. However, it is a good idea to try and test out each of the cases.



TRY IT

Directions: Run the program and test out all the Coffee drink options.

🔗 **EXAMPLE** User selects Coffee, decaf, milk, and sugar.

```

What type of drink would you like to order?
1. Water
2. Coffee
3. Tea
Drink selection #: 2
Would you like decaf? (Y/N): y
Would you like 1) milk, 2) cream, or 3) none?
Enter choice #: 1
Would you like sugar? (Y/N): y
Your drink selection: Coffee, decaf, milk, sugar.

```



REFLECT

Although this concludes the program here, there are some underlying issues with this type of code that you should be aware of. One of the issues involves the program becoming very difficult to read as it gets larger with all of the branches. In a future tutorial, you will look at cleaning up when it comes to creating more complex methods. Another issue arises when an incorrect input is entered at any point. The program is not prompting

the user to enter in a value a second time (or even a third). You can use loops to address this (this will be discussed in a future tutorial).

Since this program is finished to this point, it does consist of more selection statements than anything previously. This would be a good program to get a better understanding of the use of some extra, temporary `println()` statements to help check the values of variables and monitor flow through the program.



SUMMARY

In this lesson, you were given a project request from a company to build a drink order program by **planning algorithms**. You considered the drink order options and did some preliminary algorithm planning. Based on the drink options required, you used various selection statements to move the user through the options. Finally, you **wrote the code** for a program that, when executed, would provide the user with some input choices and an output of their final selection.

Source: This content and supplemental material has been adapted from Java, Java, Java: Object-Oriented Problem Solving. Source cs.trincoll.edu/~ram/jjj/jjj-os-20170625.pdf

It has also been adapted from “Python for Everybody” By Dr. Charles R. Severance. Source py4e.com/html3/