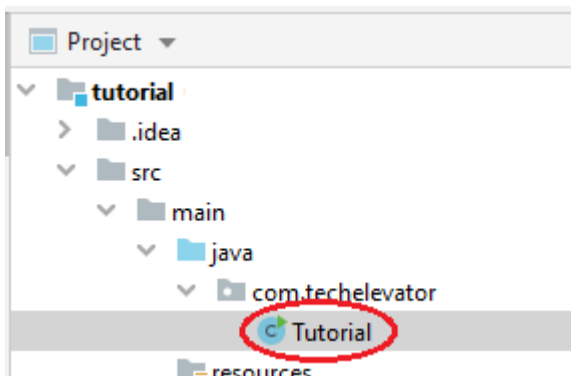


Variables and data types tutorial

In this tutorial, you'll practice determining variable data types and declaring variables using a small programming problem. You'll build a restaurant calculator, and use arithmetic expressions to calculate values and assign those values to variables.

Getting started

Open the project in IntelliJ, and in the project window, navigate to the folder `tutorial/src/main/java/com.techelevator`. You'll do all of your work in the file `Tutorial.java`.



Requirements

Before writing any code, you must first understand the overall goal of the program and its detailed *requirements*. There are many ways to express requirements for complex systems, but for small projects, a bulleted list of items suffices.

The goal of this program is to provide a tool to use at a restaurant to calculate the tax and tip given the total cost of dinner, add the tax and tip to the cost of the dinner for a grand total, and then split the grand total among a group of people.

The requirements of this program are:

- Given the total cost of dinner, calculate the tax.
- Given a tip percentage to leave to the server, calculate the tip amount based upon the total cost of dinner.
- The grand total is calculated by adding the tax and tip to the total cost of dinner.
- Given the number of people who ordered, calculate each person's share of the bill using the grand total, split evenly.
- As a bonus feature, given the total number of cookies available on the dessert tray, calculate how many cookies each guest may eat.

Step One: Declare and initialize variables

Based on the requirements, you know that you'll need the total cost of the menu items ordered, a tip percentage, and the number of guests. So, you'll need to create variables to hold these values, and initialize each variable. You must think of simple, meaningful variable names so that it's clear what your program does. These are good names:

- `costOfDinner`
- `numberOfGuests`
- `tipPercent`

Now you must think about data types:

- `costOfDinner` is numeric and can have decimal places, so make it a `double`.
- `numberOfGuests` is numeric and is a whole number, so make it an `int`.
- `tipPercent` is usually determined in whole percentages (15%, 18%, 20%), so make it an `int`.

Look at the code in the `Tutorial.java` file. Step One has been completed for you:

```
/* ***** */
// Step 1: Declare and initialize variables
/* ***** */
double costOfDinner = 120.00;
int tipPercent = 18;
int numberOfGuests = 4;

// Greet the user
System.out.println("*****");
System.out.println("*** Welcome to the Restaurant Calculator ***");
System.out.println("*****");
System.out.println("Cost of dinner: $" + costOfDinner);
```

The last four lines of this section show how you can print messages to the user's terminal. You'll learn much more about this later, but for now, understand that this is how your program "talks" to the user.

Now run the code to see how it behaves. In the window at the bottom of your screen, you can see your program output. There's a line indicating the command used to run your program, followed by your program output:

```
*****
*** Welcome to the Restaurant Calculator ***
*****
Cost of dinner: $120.0
*****
** Thanks for using Restaurant Calculator **
*****
```

Don't worry too much about the format of the number. Later, you'll learn how to print numbers in the exact format that you need, but for now, `120.0` works fine.

Step Two: Calculate the sales tax and tip

Given the cost of dinner and the tip percentage, you can calculate the amount of tip, and given a sales tax percentage, you can calculate the tax amount. These calculations are *arithmetic expressions*, and you store the results of these expressions in variables.

First, calculate the sales tax. After the **// Step 2:** comment, add these lines:

```
final double SALES_TAX_PERCENT = 7.5;
double taxAmount;
taxAmount = SALES_TAX_PERCENT / 100 * costOfDinner;
```

You defined the tax rate as a *constant*, indicated by the **final** keyword. This is because the tax percentage is a fixed number that won't change while your program runs.

Because the tax rate is a percent value, you must divide it by 100 (to get 0.075 in this case), then multiply that rate times the cost of dinner.

Note that you *declared* the variable **taxAmount** on one line, then you *assigned* its value on the next line. Declaring and assigning are distinct steps, although they're often combined into one line.

Now add this code to calculate the tip amount:

```
double tipAmount = tipPercent / 100 * costOfDinner;
```

Note that in this code, you *declared* and *assigned* **tipAmount** all in one line.

Finally, show the result to the user. Paste these print statements after the code you wrote:

```
// Display the tax and tip
System.out.println("Tax: $" + taxAmount);
System.out.println("Tip: $" + tipAmount);
```

Run your program again, and you see more output:

```
*****
*** Welcome to the Restaurant Calculator ***
*****
Cost of dinner: $120.0
Tax: $9.0
Tip: $0.0
```

But there's a problem here: the tip amount is \$0. There's a bug somewhere in this code.

Take a look at this line:

```
double tipAmount = tipPercent / 100 * costOfDinner;
```

Recall that `tipPercent` is type `int`. The literal value `100` is also `int`. When Java divides two `int` values, the result is `int`. Java drops the decimal portion of the result. So instead of the expected `0.18` as a result of this division, Java gives you `0`. Then Java multiplies that intermediate result by `costOfDinner` (a `double`), resulting in `0.0`.

There are many ways to fix this issue. You could change `tipPercent` to type `double`, or you can change the other operand, `100`, to type `double`.

Replace that line of code with the following by adding `.0` after `100`:

```
double tipAmount = tipPercent / 100.0 * costOfDinner;
```

Now when you run the program, you see this:

```
*****
*** Welcome to the Restaurant Calculator ***
*****
Cost of dinner: $120.0
Tax: $9.0
Tip: $21.599999999999998
```

The program now calculates the tip properly. Remember that due to the imprecision of decimals, you can sometimes see formatting issues as you see here. Don't worry about this for now—you'll learn how to take care of these later.

Step Three: Calculate the amount per person

Now calculate the per-guest share by adding the cost of dinner, tax, and tip, and then dividing by the number of guests.

After the **// Step 3:** comment, add these lines:

```
double amountPerPerson = costOfDinner + taxAmount + tipAmount / numberOfGuests;
// Display the amount per person
System.out.println("Amount per person: $" + amountPerPerson);
```

Run the program and view the results:

```
*****
*** Welcome to the Restaurant Calculator ***
*****
Cost of dinner: $120.0
Tax: $9.0
Tip: $21.599999999999998
Amount per person: $134.4
```

But \$120 plus tax and tip divided among four people doesn't equal \$134.40. It looks like there's another bug to address.

Look again at this line:

```
double amountPerPerson = costOfDinner + taxAmount + tipAmount / numberOfGuests;
```

Remember the concept of operator precedence in arithmetic expressions. Multiplication and division have a *higher precedence* than addition and subtraction. This means that Java first performs the division of `tipAmount / numberOfGuests`, then adds that result to `costOfDinner + taxAmount`.

What you want is to divide *the sum of dinner, tax, and tip* by the number of guests. You use parentheses to indicate which expressions to evaluate first.

Replace this calculation with one that uses parentheses to override the default operator precedence:

```
double amountPerPerson = (costOfDinner + taxAmount + tipAmount) / numberOfGuests;
```

Run the program, and see the correct result:

```
*****
*** Welcome to the Restaurant Calculator ***
*****
Cost of dinner: $120.0
Tax: $9.0
Tip: $21.599999999999998
Amount per person: $37.65
```

Step Four: Determine how much dessert each guest gets

The meal is over, and the guests receive some sort of after-dinner treats to share, such as fortune cookies or mints. Now, you need to add code to determine how many cookies each guest gets to eat and the number left over after each guest has eaten.

Under the **Step 4:** comment, add this code:

```
// Declare and initialize the number of dessert pieces
int numberOfCookies = 9;
int numberCookiesPerGuest = numberOfCookies / numberOfGuests;
int leftoverPieces = numberOfCookies % numberOfGuests;
System.out.println("Each guest can eat " + numberCookiesPerGuest + " cookies, with "
+ leftoverPieces + " left over.");
```

After you declare and initialize the number of cookies on the tray, you calculate the number of pieces per guest by dividing. Since you're doing `int` division of `numberCookies / numberOfGuests`, Java determines the result as `int` and truncates the decimal portion. In effect, this gives you the *whole number* portion of the result, which is exactly what you want in this case.

To determine how many cookies remain, use the *remainder* operator `%`. The result of this operation is the `int` remainder left after the `int` division.

When you run your program again, you see output similar to the following:

```
*****
*** Welcome to the Restaurant Calculator ***
*****
Cost of dinner: $120.0
Tax: $9.0
Tip: $21.599999999999998
Amount per person: $37.65
Each guest can eat 2 cookies, with 1 left over.
*****
** Thanks for using Restaurant Calculator **
*****
```

Next steps

If you want to experiment more, go back to the starting code in Step 1, and change values for the cost of dinner, tip percentage, and number of guests. After each change, run the program, and see how your program output changes based on the starting data.