

Explore Data Types

Estimated time needed: 20 minutes

In this lab, you will learn about the different data types that are available for you to use within a Java program.

You are currently viewing this lab in a Cloud based Integrated Development Environment (Cloud IDE). It is a fully-online integrated development environment that is pre-installed with JDK 21, allowing you to code, develop, and learn in one location.

Learning Objectives

After completing this lab, you will be able to:

- Understand what primitive data types are
- Create variable of different data types and use them in the java program
- Convert data from one type to another.

Primitive data types

Java supports the following primitive data types:

- byte
- short
- int
- long
- char
- boolean
- float
- double

The ones that are used in most programs of these are int, char, boolean, float and double. You will notice that there are no data types to handle words with more than one character. This is called String and Java handles String as an array of characters. You will explore each of these, in this lab.

1. Create a project directory by running the following command.

```
mkdir my_datatypes_proj
```

2. Run the following code to create the directory structure.

```
mkdir -p my_datatypes_proj/src
mkdir -p my_datatypes_proj/classes
mkdir -p my_datatypes_proj/test
cd my_datatypes_proj
```

3. Now create a file named `DataTypes.java` inside the `src` directory.

```
touch /home/project/my_datatypes_proj/src/DataTypes.java
```

4. Click on the button below to open the file for editing.

Open **DataTypes.java** in IDE

5. Read each statement in the following program and understand what each of the data types represent. Paste the following content in `DataTypes.java`.

```
public class DataTypes {
    public static void main(String s[]) {
        //byte ranges from -128 to 127
        //use for handling file data in binary format
        byte b = 125;
        System.out.println("Byte value " + b);
        //short ranges from -32,768 to 32,767
        //Suitable for memory-constrained environments where integers are too large
        // Used for Game scores, sensor readings etc.,
        short points = 3000;
        System.out.println("Points scored as 'short' value " + points);
        //int ranges from -2^31 to (2^31)-1
        //Suitable for all general purpose use of numbers
        // Suits values that are whole numbers
        int population = 4358000;
        System.out.println("Population as 'int' value " + population);
        //long ranges from -2^63 to (2^63)-1
        //Used for large numeric values, timestamps in milliseconds
        // It has to end with an l indicating it is a long integer
        long age_in_milliseconds = 788923800000l;
        System.out.println("Age of 25 years in milliseconds as 'long' value " + age_in_milliseconds);
        //char is used to store on character
        // Suitable for capturing keyboard input one key at a time
        // Used mostly as a group called arrays for words
        char keyboard_input = 'a';
        System.out.println("Key board input is "+keyboard_input);
        //boolean is used to true or false
        // Suitable for anything where the values can only be true or false
        boolean isLabCompleted = false;
        System.out.println("The lab is completed "+isLabCompleted);
        //float is used for numbers with decimals, with precision up to 7 decimal digits
        // Suitable for representing money, measurement of area, volume, etc.,
        // It has to end with an f
        float house_value = 4000000.58f;
        System.out.println("The value of the house is $" + house_value);
        //double is used for numbers with decimal, with precision up to 15 decimal digits
        // Suitable for representing statistical, astronimcal measurements
        double pi = 3.14159265359;
        System.out.println("The value of pi is "+pi);
    }
}
```

```
}
```

5. Compile the java program, this time explicitly specifying the destination directory as the `classes` directory that you created.

```
javac -d classes src/DataTypes.java
```

6. Set the `CLASSPATH` variable.

```
export CLASSPATH=$CLASSPATH:/home/project/my_datatypes_proj/classes
```

7. Now when you run the java program, it will run seamlessly as expected.

```
java DataTypes
```

You will see the output as below:

```
Byte value 125
Points scored as 'short' value 3000
Population as 'int' value 4358000
Age of 25 years in milliseconds as 'long' value 788923800000
Key board input is a
The lab is completed false
```

The value of the house is \$4000000.5
The value of pi is 3.14159265359

Data type conversion from double to int

In this part of the lab, you will see how to convert a double to an int. This is useful when you want to discard the decimal part. This process of converting is called casting.

1. Now create a file named `DataTypes.java` inside the `src` directory.

```
touch /home/project/my_datatypes_proj/src/DataDoubleToInt.java
```

2. Click on the button below to open the file for editing.

Open **DataDoubleToInt.java** in IDE

3. Read each statement in the following program and understand what each of the data types represent. Paste the following content in `DataDoubleToInt.java`.

```
public class DataDoubleToInt {  
    public static void main(String s[]) {  
        double area_circle = 78.53981634d;  
        System.out.println("Area of circle of 5cm radius is " + area_circle);  
        System.out.println("Area of circle of 5cm radius is " + (int)area_circle);  
    }  
}
```

4. Compile the java program, this time explicitly specifying the destination directory as the `classes` directory that you created.

```
javac -d classes src/DataDoubleToInt.java
```

5. Now run the java program.

```
java DataDoubleToInt
```

You will see the output as below:

```
Area of circle of 5cm radius is 78.53981634
Area of circle of 5cm radius is 78
```

Practice Exercise

1. Create a file in the `src` folder that will declare a char with value 'a'.
2. Print the value of the char as is.
3. Try to cast it to an `(int)`.
4. Print the value of the converted int.
5. Compile and run it.

The number output you get is the ASCII value of the char.

► [Click here for sample code](#)

Conclusion

In this lab, you learned how to handle primitive data types through variables.

Author(s)

[Lavanya](#)