Computer Programing in Java

Lecture 2
Variables, Operators

م.م حسين عجام

Java Variables

Variables are containers for storing data values.

In Java, there are different types of variables, for example:

- **String - stores text, such as "Hello". String values are surrounded by double quotes**
- **int - stores integers (whole numbers), without decimals, such as 123 or -123**
- **float - stores floating point numbers, with decimals, such as 19.99 or -19.99**
- **char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes**
- **boolean - stores values with two states: true or false**

Declaring (Creating) Variables

- To create a variable, you must specify the type and assign it a value:

type variableName = value;

- Where type is one of Java's types (such as int or String), and variableName is the name of the variable (such as x or name). The equal sign is used to assign values to the variable.

To create a variable that should store text, look at the following example:

Example

Create a variable called name of type String and assign it the value "John":

```
String name = "John";
System.out.println(name);
```

Declaring (Creating) Variables

- **Example**

Create a variable called myNum of type int and assign it the value 15:

```
int myNum = 15;
System.out.println(myNum);
```

- You can also declare a variable without assigning the value, and assign the value later:

```
int myNum;
myNum = 15;
System.out.println(myNum);
```

- Note that if you assign a new value to an existing variable, it will overwrite the previous value:

Example

Change the value of myNum from 15 to 20:

```
int myNum = 15;
myNum = 20;  // myNum is now 20
System.out.println(myNum);
```

Final Variables

- If you don't want others (or yourself) to overwrite existing values, use the final keyword (this will declare the variable as "final" or "constant", which means unchangeable and read-only):

- **Example**

```
final int myNum = 15;
myNum = 20;  // will generate an error: cannot assign a value to a final variable
```

Other Types

- A demonstration of how to declare variables of other types:

Example

```
int myNum = 5;
float myFloatNum = 5.99f;
char myLetter = 'D';
boolean myBool = true;
String myText = "Hello";
```

Print Variables

- Display Variables
- The println() method is often used to display variables.

- To combine both text and a variable, use the + character:

Example

```
String name = "John";
System.out.println("Hello " + name);
```

You can also use the + character to add a variable to another variable:

```
String firstName = "John ";
String lastName = "Doe";
String fullName = firstName + lastName;
System.out.println(fullName);
```

- For numeric values, the + character works as a mathematical operator (notice that we use int (integer) variables here):

Example

```
int x = 5;
int y = 6;
System.out.println(x + y); // Print the value of x + y
```

Declare Multiple Variables

- Example
- Instead of writing:

```
int x = 5;
int y = 6;
int z = 50;
System.out.println(x + y + z);
```

- You can simply write:

```
int x = 5, y = 6, z = 50;
System.out.println(x + y + z);
```

Java Operators

- Java Operators
- Operators are used to perform operations on variables and values.

- In the example below, we use the + operator to add together two values:

```
int x = 100 + 50;
```

Arithmetic Operators

Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	++x
--	Decrement	Decreases the value of a variable by 1	--x