INF1512

JAVA Programming:
Data Types, Identifiers and Operators

Dr. KIMBI X.Y

How Do Programs Work and How can you develop Them?

(Java concepts – syntax and semantics of Java programming)

Exercise One (TD)

(1) Integrated Development Environments such as NetBeans often use **syntax coloring,** which assigns various colors to the characters in a program to reflect the syntax of the language.

A student notices that NetBeans colors the word ***String*** differently from **int, double,** and **boolean**. The student asks why ***String*** should be a different color, since all these words are names of data types.

What's the answer to the student's question?

(2) Write a Java program that request a user to print his/her **age**, **university**, **department,** the **number of year** already spent in this university using a scanner

(4) Write a Java program that that accepts a number X as an input value and then prints:

(5) Give the meaning of each of the following Java operators:

a)	++	d)	&&	e)	<=
b)	!=	e)	!	f)	>=
c)	- -	f)	||	e)	==

(3) Explain why the value of the expression 2 + 3 + "test" is the string "5test" while the value of the expression "test" + 2 + 3 is the string "test23". What is the value of "test" + 2 * 3 ?

(4) fd gg

The Big Picture

- Firstly, in other to develop a successful program in any programming language, you **MUST** define and revisit the functionalities of the system you are to develop **(Already Explained explicitly in our SCRUM methodology)**
- In Java programming, you should start by creating **various packages** for differences resources (images, Database, etc)
- When you use **images or other resources** in an application, typically you create a separate **Java Package** for the resources.
- **A package is a named collection of classes**

Structure of a Java Program

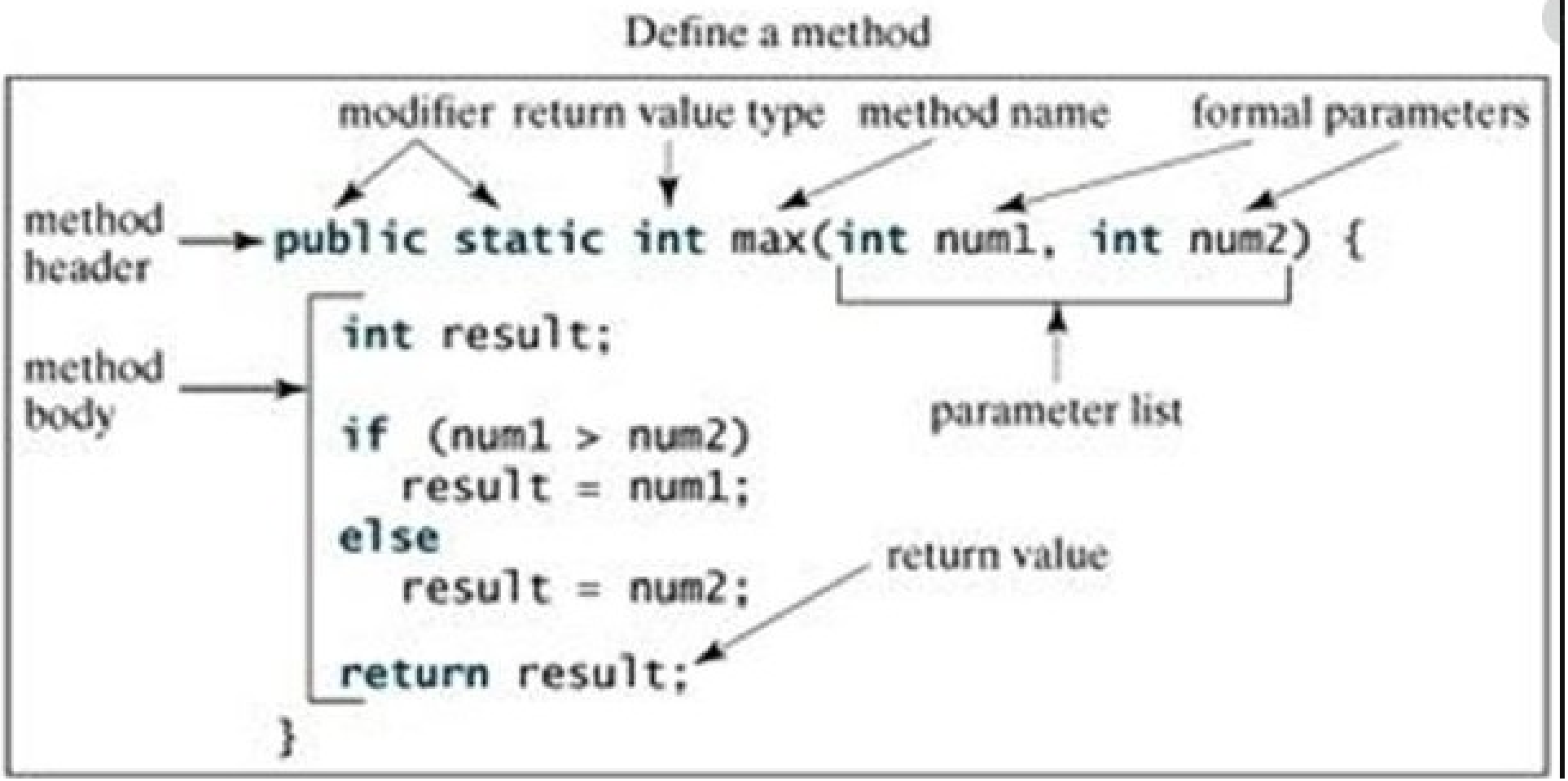
```
public class Example1 {


/* @param args */


public static void
   main(String[] args)
{
System.out.println("This
   gets printed out.");
}
}
```

- Tells compiler you're creating a class called Example1 (so the java file should be Example1.java)
- Comments
- Function/method (set of statements grouped together), called main
- Beginning of main function
- Code statement: print a line of text, end with ;
- End of main function

Structure of a Java Program



Structure of a Java Program

- All programming in Java is done inside "classes", meaning, a program MUST contain at least one class.
- The above program has the class named **Example1**
- The key word **"public" (Access Modifier)** in the first line of **main()** means that this subroutine can be called from outside the program.

In Java, access specifiers are the keywords used before a class name which defines the access scope.

- Class name must begin with a capital letter while variable names and methods should begin with a small letter. (naming convention)
-

Data Types

- Variables are nothing but reserved memory locations to store values. This means that when you create a variable, you reserved some space in the memory.
- Variables declared inside a subroutine are called local variables for that subroutine. They exist only inside the subroutine, while it is running, and are completely inaccessible from outside outside.
- There are two categories of data types (DT) in Java Programming:
  - **Primitive DT** (fundamental and built into Java)
  - **Object/Referenced DT** (User-defined)

Primitive Data Types

- Primitive data types are **predefined by the programming language** and named by a key word. There are eight primitives data types supported by Java programming(byte, short, int, long, float, double, char, boolean).
- The first four hold integer numbers while float and double hold real numbers.
- A programmer **cannot** create new primitive data types.
- Any data type you invent will be of a type **object.**
- Most commonly used in practice are int, double and float
- **bytes:** 8bits = 1byte (8-bits integer)
- **short:** 16-bits integer
- **int:** 32-bits integer
- **long:** 64-bits integer
- **float:** 32-bits floating point
- **double:** 64-bits floating point
- **char:** 16-bits Unicode character
- **boolean:** boolean variables false or true

Primitive Data Types

Java primitive data types

Primitive Type	Description	Range
`byte`	8-bit integer	-128 to 127
`short`	16-bit integer	-32768 to 32767
`int`	32-bit integer	-2147483648 to 2147483647
`long`	64-bit integer	-2^{63} to $2^{63}-1$
`float`	32-bit floating point	10^{-46} to 10^{38}
`double`	64-bit floating point	10^{-324} to 10^{308}
`char`	Unicode character	
`boolean`	Boolean variable	`false` and `true`

Referenced Data Types (Object Data Types)?

• Referenced variables are **created defined constructors** of the classes. They are used to access objects.

•Class objects and various types of arrays objects come under referenced data types.

•A referenced variable can be used to refer to any object of the declared type or any compatible type.

•Example 1: Registration reg = **new** Registration ();

•Example 2: Scanner sc = **new** Scanner (System.in);

•Example 1: Student reg = **new** Registration ();

•Example 1: Department reg = **new** Department ();

Exercise One

(1) Create a Java Class called **Welcome** and write a Java Program to print the following statements:

- You are welcome to INF1512
- Lateness is not allowed
- My name is Favour

(2) Create a Java Class called personnalInformation and write a Java Program to enter the following using a **Scanner**: Name, Department, Specialty, Faculty, University, DOB, and Gender

Assignment

Explain the concept of Data Types as used in Object-Oriented Programming with emphasis on the differences between Primitive Data Types and Non-Primitive data types .

(11) Write down the codes in Java for:

(i) Adding two numbers

(ii) Subtracting two numbers

(iii) Dividing two numbers

(iv) Multiplying two numbers

(v) Calculating the minimum of ten numbers

Java Literals

A literal is a source code representation of a fixed value. They are represented directly in the code

without any computation.

Literals can be assigned to any primitive type variable. For example:

Text Input and Output

```
System.out.print(x);
System.out.println();
Scanner sc = new Scanner(System.in);
```

Exercise one

(1) **Found Above (String is class and the value of type string is an object)**
The String class defines a lot of functions/methods:
.equals(s2) is a function that returns a boolean value.
.length() is an integer-valued function that gives the number of characters
Furthermore, You can use the plus operator, +, to concatenate two or more strings.

(2) Write a program that permits a use to enter inputs using Scanner: i.e the name of your **University** and your **department,** your **name** and then greets the user by name. Before outputting the user's name, convert it to upper case letters. For example, if the user's name is Xaveria, then the program should respond as follows:

“The name of my University is UIECC located in Sammalima”
“The Name of department is Computer Science Department”
"Hello, XAVERIA, nice to meet you!

(2) Write a java program that asks you to enter your full **name**, your **department**, your **university**, your **country of origin**, your **age**, your expected **graduation year and** your **date of birth**. After you enter them, write out each answer on a separate line. You must use Scanner to input the 7 fields, and a single println to output the 7 fields separately.

Operators: Arithmetic Operators

•Arithmetic operators include addition, subtraction, multiplication, and division. They are indicated by +, -, *, and / respectively.

•When the computer actually calculates one of these operations, the two values that it combines must be of the same type.

Exercise Two

(1) Given that the values of x and y are 10 and 3 respectively. Compute the sum, subtraction, multiplication and division. What were your observations?

•

(2a) A common use of percent (%) is to test whether a given integer is ---------- or---------- or ----------

(2b) Define the term casting as used in Java Programming

(3) Write a Java program to print out all even numbers between 1 up to 100 (4) Write a Java program to print out all odd numbers from 1 up to 100

(5) Write a Java program to print out all multiples of 3 from 1 up to 100

Operators: Logical Operators

Logical Operators return true of false. Assuming x and y are two variables:

```
x==y            // x is equal to y
x!=y            // x is not equal to y
x<y             // x is lesser than y
x<=y            // x is less than or equal to y
x>y             // x is greater than y
x==y            // x is greter than or equal to y
```

Operators: Boolean Operators

We used boolean operators for combining operators. Again, these return true or false.

x&&y //the **AND** operator (True if both x and y are true)

x||y //the **OR** operator (True if either x and y (or both) are true)

!x //the **NOT** operator **(Negation)** (True if x is false)

Example 1: (x > 0 && y > 0 && z > 0) // All theses are positives variables

Example 2: (x < 0 || y < 0 || z < 0) // At least one of the three variable is negative

These operators will be commonly used as test expressions in selection statements or repetition statements (loop).

Built-in Subroutines and Functions
Math Methods

•A class is a collection of related item. One of the purposes of a class is to group together some variables and subroutines, which are contained in that class. These variables and subroutines are called **static members** of the class. You've seen one example: In a class that defines a program, the **main()** routine is a static member of the class. The parts of a class definition that define static members are marked with the reserved word **"static"**, such as the word **"static"** in public static void main…

•You are familiar with the mathematical function that computes the square root of a number. The corresponding function in Java is called **Math.sqrt**. This function is a static member subroutine of the class named Math. If x is any numerical value, then Math.sqrt(x) computes and returns the square root of that value. Since Math.sqrt(x) represents a value, it doesn't make sense to put it on a line by itself in a subroutine call statement such as:

```
Math.sqrt(x); // This doesn't make sense!
```

•What, after all, would the computer do with the value computed by the function in this case? You have to tell the computer to do something with the value. You might tell the computer to display it:

System.out.print(Math.sqrt(x)); // Display the square root of x.

•The Math class contains many static member functions. Here is a list of some of the more important of them:

- Math.abs(x), computes the absolute value of x.
- Math.pow(x,y) for computing x raised to the power y
- Math.floor(x), which rounds x down to the nearest integer value that is less than or equal to x. Even though the return value is mathematically an integer, it is returned as a value of type double, rather than of type int as you might expect. For example, Math.floor(3.76) is 3.0. The function Math.round(x) returns the integer that is closest to x, and Math.ceil(x) rounds x up to an integer. (“Ceil” is short for “ceiling”, the opposite of “floor.”)
- Math.min (a, b), compute the minimum of a and b.
- Math.max (a, b), compute the maximum of a and b.
- Math.randon(), compute random numbers
-

How to generate radom numbers

Random numbers within a specific range of type integer, float, double, long, and boolean can be generated in Java.

There are three methods to generate random numbers in Java:

Method 1: Using random class

To use the Random Class to generate random numbers, follow the steps below:

(1) Import the class java.util.Random
(2) Make the instance of the class Random, i.e., Random rand = new Random()
(3) Invoke one of the following methods of rand object:
nextInt(upperbound) generates random numbers in the range 0 to upperbound-1.
nextFloat() generates a float between 0.0 and 1.0.
nextDouble() generates a double between 0.0 and 1.0.

Run

Method 2: Using Math.random

For generating random numbers within a range using Math.random(), follow the steps below:

Declare the minimum value of the range
Declare the maximum value of the range
Use the formula Math.floor(Math.random()*(max-min+1)+min) to generate values with the min and the max value inclusive.
Note: This method can only be used if you need an integer or float random value

```java
public class Random {
    public static void main(String[] args) {
        int max = 20; int min = 10;
       int range = max - min + 1;
       int num = (int)(range * Math.random() + min);
        System.out.println("Random mumber: " +num);


    }
}
```

Method 3: ThreadLocalRandom

To generate random numbers using the class **ThreadLocalRandom,** follow the steps below:

(1) Import the class java.util.concurrent.ThreadLocalRandom

(2) Call the following method

To generate random number of type int:
ThreadLocalRandom.current().nextInt()

To generate random number of type double:
ThreadLocalRandom.current().nextDouble()

To generate random number of type boolean:
ThreadLocalRandom.current().nextBoolean()