Department of Software Engineering
Mehran University of Engineering and Technology, Jamshoro

Course: SWE121 – Object Oriented Programming			
Instructor	Mariam Memon	**Practical/Lab No.**	01
Date		**CLOs**	3
Signature		**Assessment Score**	

Topic	**Installation of Java, Basic Program Structure, Identifiers and Primitive data types**
Objectives	To configure Java software developers kit, Java run time environment, Java virtual machine. To sketch, compile and run basic java programs.

Lab Discussion: Theoretical concepts and Procedural steps

Java Development Kit

A software development environment for developing Java applications. Java Development Kit (JDK) is comprised of the following basic components:

- Java compiler (javac)
- Java Virtual Machine (JVM)
- Java Application Programming Interface (API)
- JRE Java Runtime Environment

Java Compiler

The name of the Java compile is Javac. The Java Language Compiler compiles programs written in the Java Programming language and translates them into platform independent Bytecodes. Produces a .class file containing bytecodes.

```
javac MyProgram.java
```

Java Virtual Machine

Java Virtual Machine (JVM) is the engine that executes java bytecodes. It is a software that converts the .class file containing bytecodes into platform specific machine code such as windows, linux, solaris etc. java command is used to launch the JVM.

Java Core API Or Java Class libraries

A library in Java is a collection of classes. The classes are grouped together into related sets that are called packages, like
java.lang (Object, Integer, String ….)

java.util	(Vector, Hashtable, Date....)
java.awt	(Window, Menu, Button....)
java.net	(Socket, URL, DatagramPacket....)

https://docs.oracle.com/javase/7/docs/api/

Java Runtime Environment

As the runtime environment for Java, the JRE contains the Java class libraries, the Java class loader, and the Java Virtual Machine. In this system:

- The *class loader* is responsible for correctly loading classes and connecting them with the core Java class libraries.
- The *JVM* is responsible for ensuring Java applications have the resources they need to run and perform well in your device or cloud environment.
- The *JRE* is mainly a container for those other components, and is responsible for orchestrating their activities.

Installing Java

The Java Development Kit (JDK), officially named "Java Platform Standard Edition" or "Java SE", is needed for writing Java programs. The JDK is freely available from Sun Microsystems (now part of Oracle). The mother site for JDK (Java SE) is http://www.oracle.com/technetwork/java/javase/overview/index.html.

Step 1: Download JDK

1. Goto Java SE download site @ http://www.oracle.com/technetwork/java/javase/downloads/index.html.
2. Under "Java Platform, Standard Edition" ⇒ "Java SE 11.0.{x}", where {x} denotes a fast running security-update number ⇒ Click the "Oracle JDK" "Download" button.
3. Under "Java SE Development Kit 11.0.{x}" ⇒ Check "Accept License Agreement".
4. Choose the JDK for your operating system, i.e., "Windows". Download the "**exe**" installer (e.g., "`jdk-11.0.{x}_windows-x64_bin.`**`exe`**" - about 150MB).

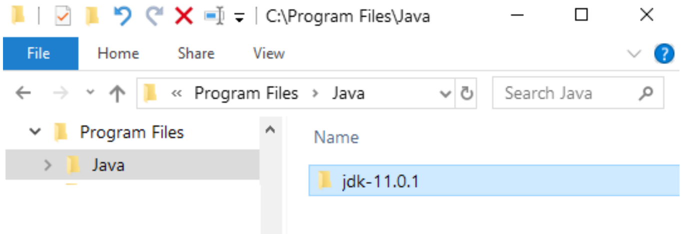
Step 2: Install JDK

Run the downloaded installer (e.g., "`jdk-11.0.{x}_windows-x64_bin.exe`"), which installs both the JDK and JRE.

By default, JDK is installed in directory "`C:\Program Files\Java\jdk-11.0.{x}`", where `{x}` denotes the update number. Accept the defaults and follow the screen instructions to install JDK.

Use your "File Explorer", navigate to "`C:\Program Files\Java`" to inspect the sub-directories. Take note of your **JDK installed directory**, in particular, the update number `{x}`, which you will need in the next step.

In the following diagram, the JDK installed directory is "`C:\Program Files\Java\jdk-11.0.1`", where `{x}=1`.



jdk-11.0.1

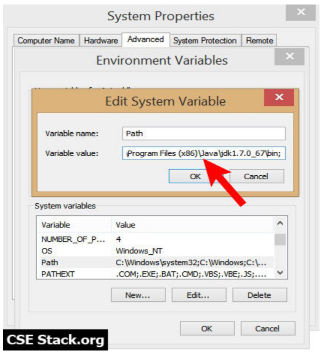
Step 3: Include JDK's "bin" Directory in the PATH

Windows' Command Prompt (`CMD`) searches the current directory and the directories listed in the `PATH` *environment variable* (or *system variable*) for executable programs. JDK's programs (such as Java compiler "`javac.exe`" and Java runtime "`java.exe`") reside in the *sub-directory* "`bin`" of the JDK installed directory. You need to include JDK's "`bin`" in the `PATH` to run the JDK programs.

To edit the `PATH` environment variable in Windows 10:

1. Launch "Control Panel" ⇒ (Optional) "System and Security" ⇒ "System" ⇒ Click "Advanced system settings" on the left pane.
2. Switch to "Advanced" tab ⇒ Click "Environment Variables" button.
3. Under "System Variables" (the bottom pane), scroll down to select variable "Path" ⇒ Click "Edit...".

4. Add the JDK bin directory to the path.



Variable name: Path

Variable value: Program Files (x86)\Java\jdk1.7.0_67\bin;

Variable	Value
NUMBER_OF_P...	4
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

5. Under "System Variables" (the bottom pane) ⇒ Click "New" (or Look for "JAVA_HOME" and "Edit" if it is already set) ⇒ In "Variable Name", enter "JAVA_HOME" ⇒ In "Variable Value", enter your JDK installed directory

Step 4: Verify the JDK Installation

Launch a CMD via one of the following means:

1. Click "Search" button ⇒ Enter "cmd" ⇒ Choose "Command Prompt", or
2. Right-click "Start" button ⇒ run... ⇒ enter "cmd", or
3. Click "Start" button ⇒ Windows System ⇒ Command Prompt

Issue the following commands to verify your JDK installation:

1. Issue "path" command to list the contents of the PATH environment variable. Check to make sure that your JDK's "bin" is listed in the PATH.

   ```
   path
   PATH=c:\Program Files\Java\jdk-11.0.{x}\bin;[other entries...]
   ```

2. Issue the following commands to verify that JDK/JRE are properly installed and display their version:

   ```
   // Display the JDK version
   javac -version
   javac 11.0.1

   // Display the JRE version
   java -version
   java version "11.0.1" 2018-10-16 LTS
   Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)
   Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS,
   mixed mode)
   ```

Basic Java Program

Step 1) Open any Text Editor of your choice like notepad, textpad, notepad++, etc

Step 2) Create the Source Code for your Program

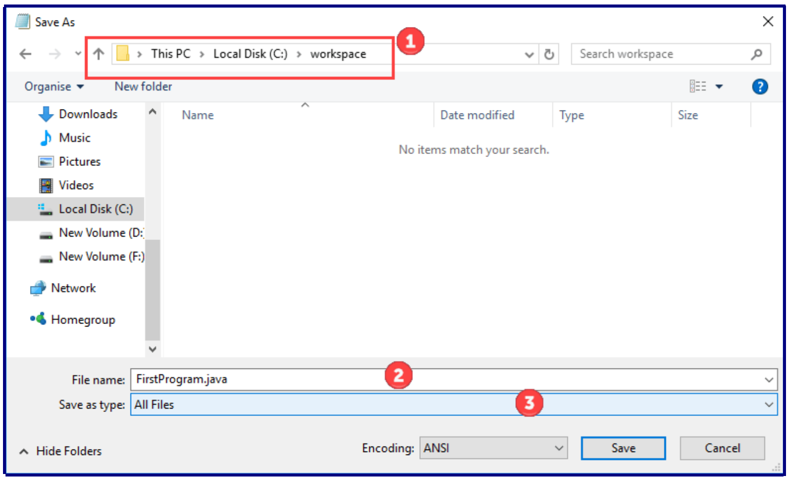
- Declare a class with name FirstProgram.java
- Declare the main method **public static void main(String args[]){**
- Now Type the **System.out.println("Hello World");** which displays the text Hello World.

```
class FirstProgram{

    public static void main(String args[]){
```

```
            System.out.println("Hello World");
        }
}
```

Step 3) Save the file as **FirstProgram.java** make sure to select file type as all files while saving the file in our working folder **C:\ workspace**



This PC > Local Disk (C:) > workspace

No items match your search.

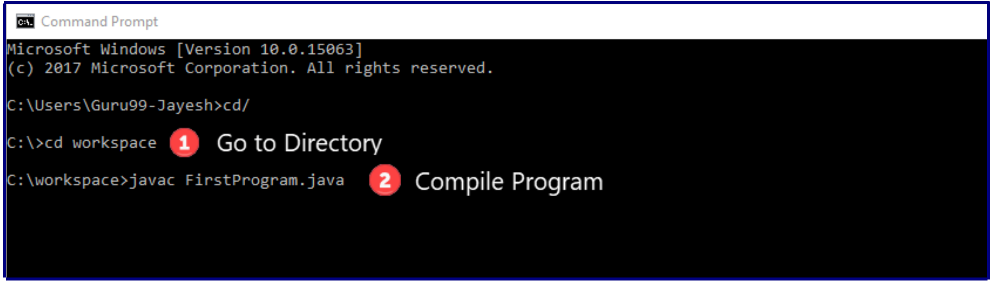
File name: FirstProgram.java

Save as type: All Files

Encoding: ANSI

Step 4) Open the command prompt. Go to Directory **C:\ workspace**. Compile the code using command,

```
javac HelloWorld.java
```



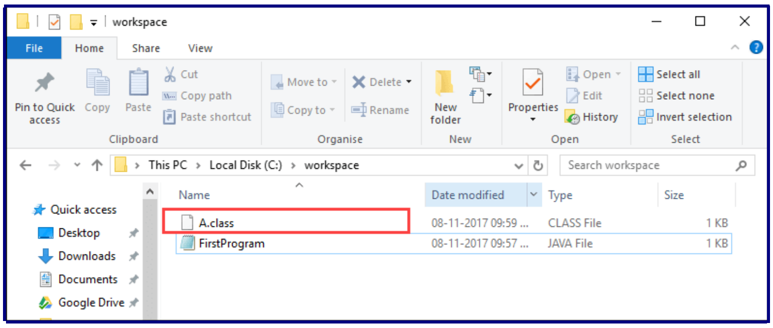
```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Guru99-Jayesh>cd/

C:\>cd workspace

C:\workspace>javac FirstProgram.java
```

Step 5) If you look in your working folder, you can see that a file named **FirstProgram.class** has been created.



This PC > Local Disk (C:) > workspace

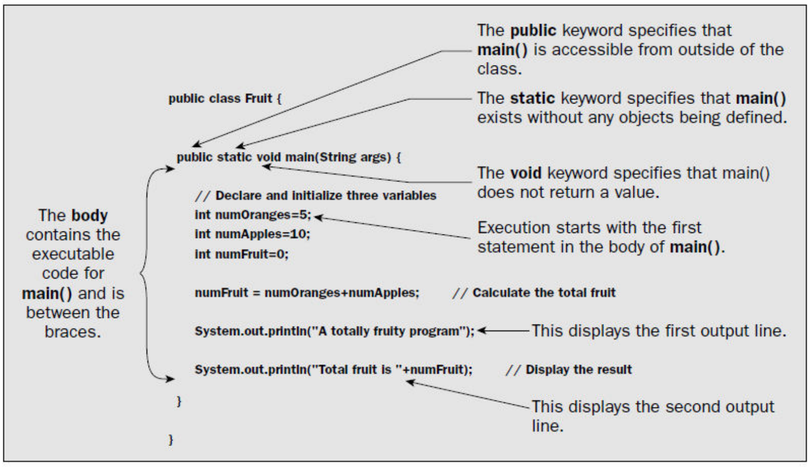
Name	Date modified	Type	Size
A.class	08-11-2017 09:59 ...	CLASS File	1 KB
FirstProgram	08-11-2017 09:57 ...	JAVA File	1 KB

Step 6) To execute the code, enter the command java followed by the class name, as expected output **Hello World** is displayed now.

```
java FirstProgram
```

Note: Java is case sensitive Programming language. All code, commands, and file names should is used in consistent casing. **FirstProgram** is not same as **firstprogram.**

Basic Program Structure



```
public class Fruit {

 public static void main(String args) {

   // Declare and initialize three variables
   int numOranges=5;
   int numApples=10;
   int numFruit=0;

   numFruit = numOranges+numApples;      // Calculate the total fruit

   System.out.println("A totally fruity program");

   System.out.println("Total fruit is "+numFruit);      // Display the result
 }

}
```

The **public** keyword specifies that **main()** is accessible from outside of the class.

The **static** keyword specifies that **main()** exists without any objects being defined.

The **void** keyword specifies that main() does not return a value.

Execution starts with the first statement in the body of **main()**.

This displays the first output line.

This displays the second output line.

The **body** contains the executable code for **main()** and is between the braces.

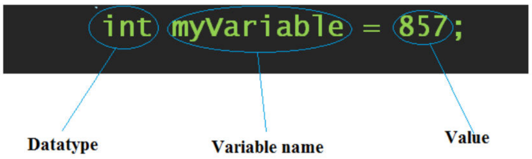
Identifiers

Identifiers are used for class names, method names, and variable names. Java is case-sensitive so VAL and val are two different identifiers.

Valid identifiers	Invalid Identifiers
AvgTime	2count
count	high-temp
a4	Not/ok
$test	
this_is_ok	

Variables

A variable is a program component used to store or represent data. The number, letter, or other data item in a variable is called its value. The value for a variable can be changed in the program. Each variable that you declare can store values only of a type consistent with the data type of that variable.

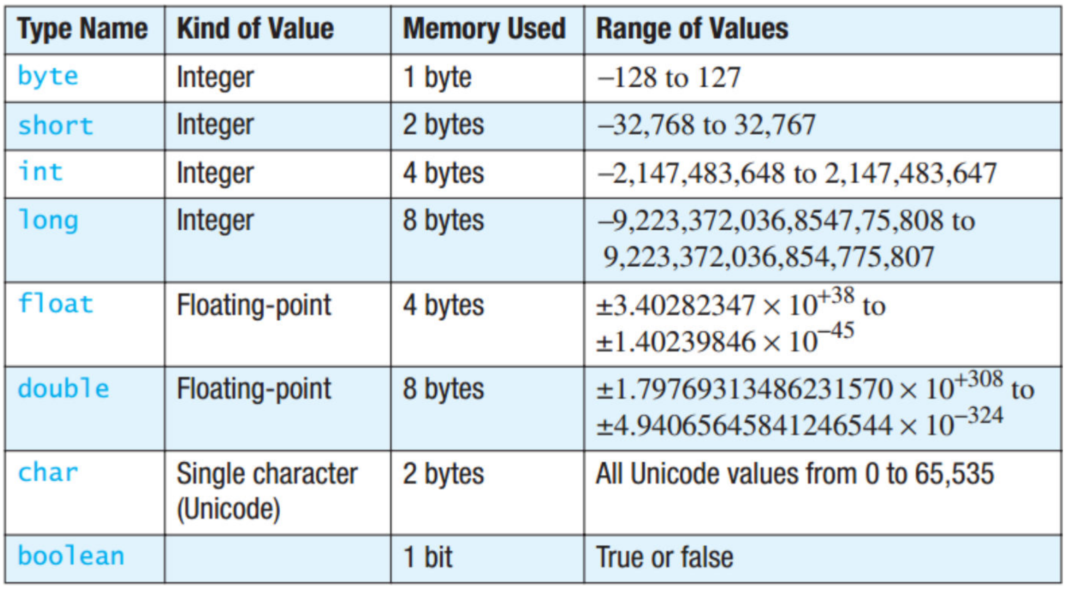


```
int myVariable = 857;
```

Data Types

The only things in Java that are not objects are data types. The fundamental types are referred to as primitive types. The values can fall in one of the following category:

- Numeric values, which can be either integer or floating-point
- A single Unicode character
- Logical values that can be true or false



Type Name	Kind of Value	Memory Used	Range of Values
byte	Integer	1 byte	–128 to 127
short	Integer	2 bytes	–32,768 to 32,767
int	Integer	4 bytes	–2,147,483,648 to 2,147,483,647
long	Integer	8 bytes	–9,223,372,036,8547,75,808 to 9,223,372,036,854,775,807
float	Floating-point	4 bytes	±3.40282347 × 10^{+38} to ±1.40239846 × 10^{-45}
double	Floating-point	8 bytes	±1.79769313486231570 × 10^{+308} to ±4.94065645841246544 × 10^{-324}
char	Single character (Unicode)	2 bytes	All Unicode values from 0 to 65,535
boolean		1 bit	True or false

Type Conversion and Casting

Type Casting Assigning a value of one type to a variable of another type is known as Type Casting.

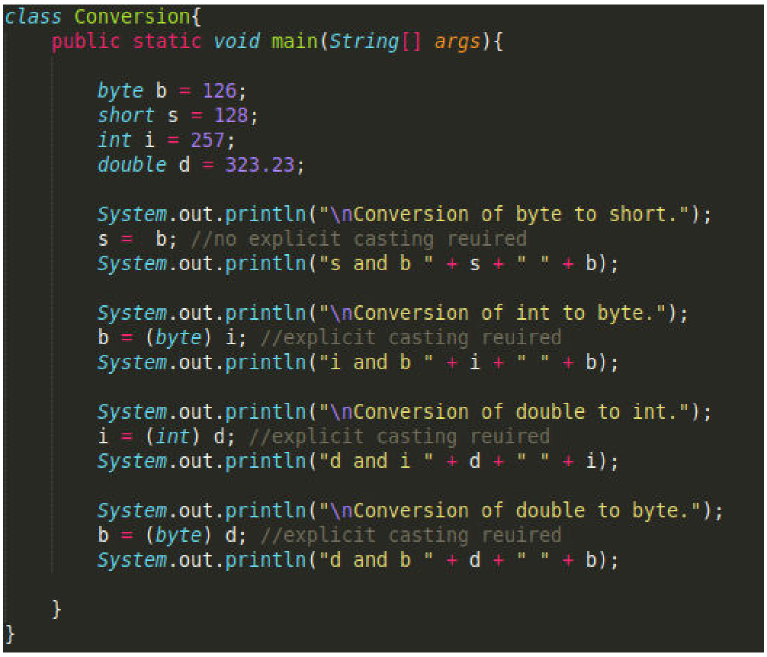
Example

```
int  x = 10;
byte  y = (byte) x;
```

In Java, type casting is classified into two types:

Widening Casting(Implicit) Conversion: It takes place when two types are compatible and the destination type is larger than the source type.
E.g. byte value assigned to int type.

Narrowing Casting(Explicitly done): Narrowing conversion takes place when a value of larger data type is assigned to the variable of smaller data type.
E.g. An int value to a byte variable.



```
class Conversion{
    public static void main(String[] args){

        byte b = 126;
        short s = 128;
        int i = 257;
        double d = 323.23;

        System.out.println("\nConversion of byte to short.");
        s =  b; //no explicit casting reuired
        System.out.println("s and b " + s + " " + b);

        System.out.println("\nConversion of int to byte.");
        b = (byte) i; //explicit casting reuired
        System.out.println("i and b " + i + " " + b);

        System.out.println("\nConversion of double to int.");
        i = (int) d; //explicit casting reuired
        System.out.println("d and i " + d + " " + i);

        System.out.println("\nConversion of double to byte.");
        b = (byte) d; //explicit casting reuired
        System.out.println("d and b " + d + " " + b);

    }
}
```

Lab Tasks

Task 1: Write a program in which declare some variables with valid identifiers, to hold your name, your total marks in previous semester, percentage, your grade, your status of pass or fail (by a Boolean variable) etc, assign them explicitly and print them. Try to declare variables of all (8) data types and assign the appropriate values.

Task 2: Write a program to declare and initialize a double variable with some value such as 50.25. Then retrieve the integral part and store it in the variable of type long, and the fractional part and store it in a variable of type double. Display the actual number, the integral part and the Fractional part.