Object Oriented Programming Practicum

ICT2132

Introduction to Java

P.H.P. Nuwan Laksiri
Department of ICT
Faculty of Technology
University of Ruhuna

Lesson 01

Course Plan

- Practicals – 04 Hours per Week
- Evaluation
  - Mid Term Evaluation – individual Practical
  - Mini project - Group

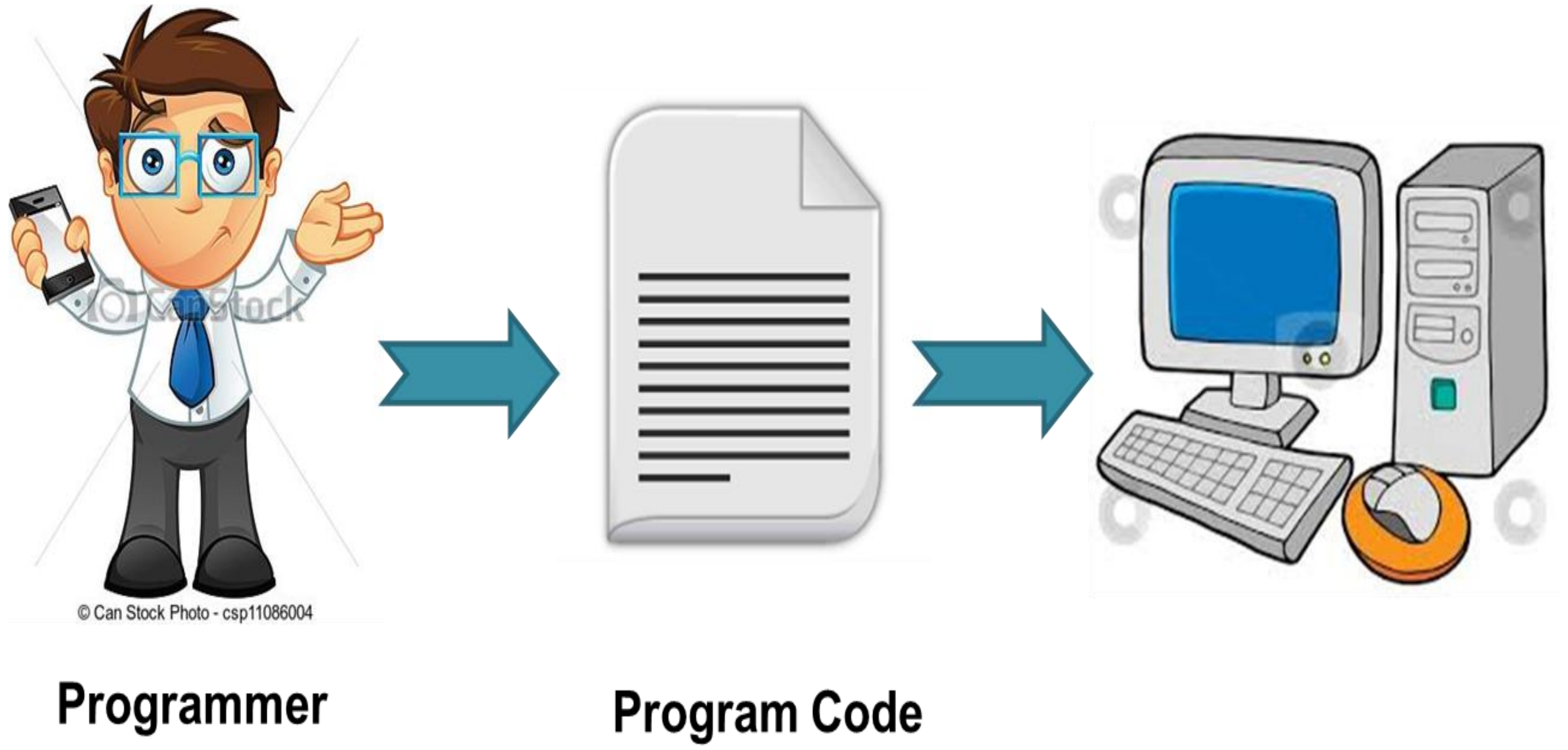
Eligibility and Evaluation Criteria

- Eligibility
  - **80%** Attendance is **MANDATORY**
  - **50%** from CA is **MANDATORY**
- Evaluation Criteria
  - CA (40%)
    - 20% Mid Term Practical Evaluation
    - 20% Mini project
  - ESA (60%)
    - 60% From Final Practical Exam

Outline

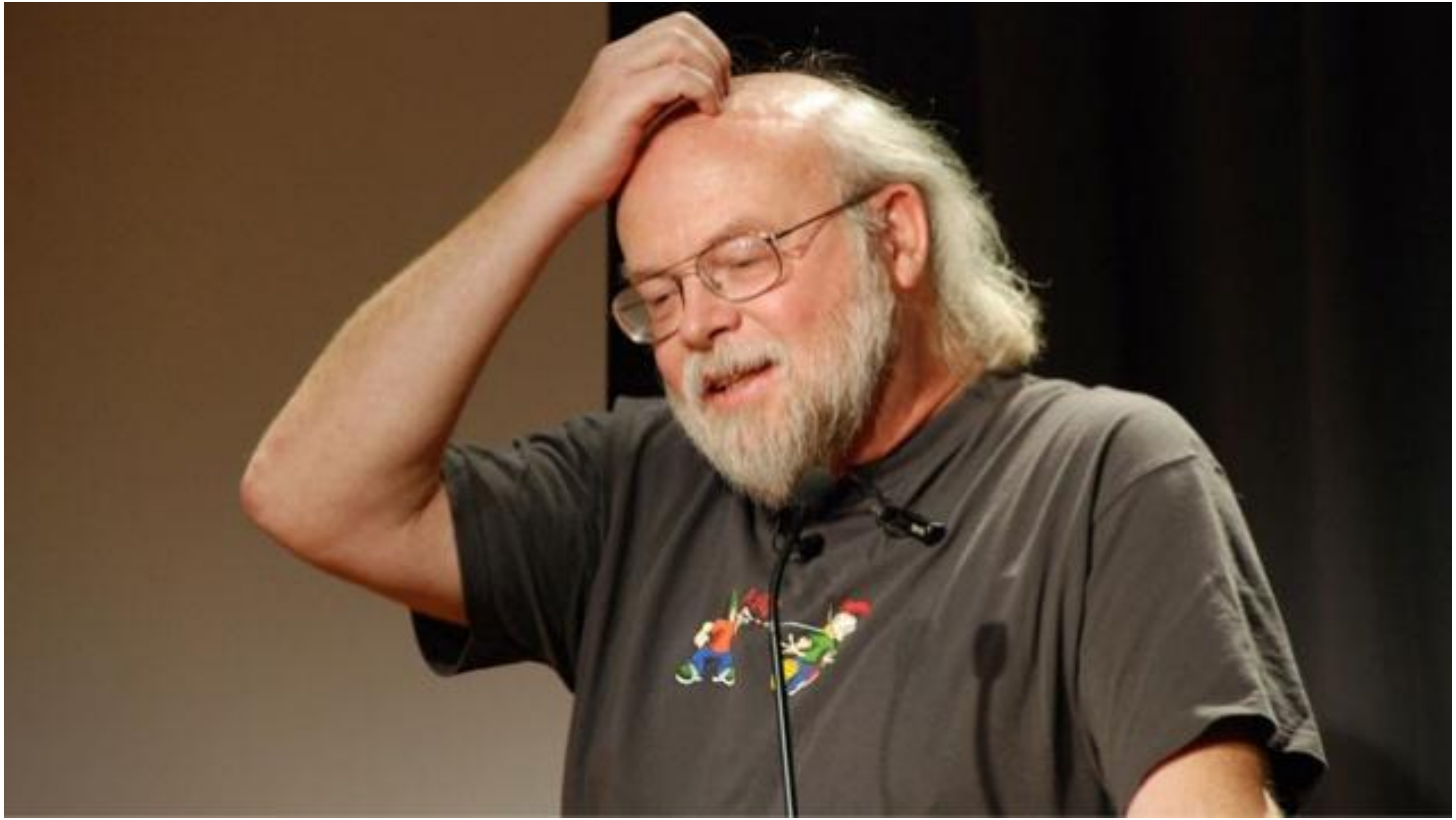
What is Programming?

- The process of writing computer programs



What is Object Oriented Programming

- Object-oriented programming
  - is a paradigm
  - based on the concept of wrapping pieces of data, and behavior related to that data,
  - into special bundles called objects,
  - which are constructed from a set of "blueprints",
  - defined by a programmer,
  - called classes.



Java™

ORACLE

JAVA - History

- A programming language created by James Gosling from Sun Microsystems (Sun) in 1991.
- First version of Java (Java 1.0) was released in 1995
- Sun Microsystems was acquired by the Oracle Corporation in 2010
- The current version of Java SE is 19(2022)
- In 2006, Java was made available under GNU General Public License
- There were five primary goals in the creation of the Java language:
  - It must be "simple, object-oriented, and familiar".
  - It must be "robust and secure".
  - It must be "architecture-neutral and portable".
  - It must execute with "high performance".
  - It must be "interpreted, threaded, and dynamic".

Platforms of Java

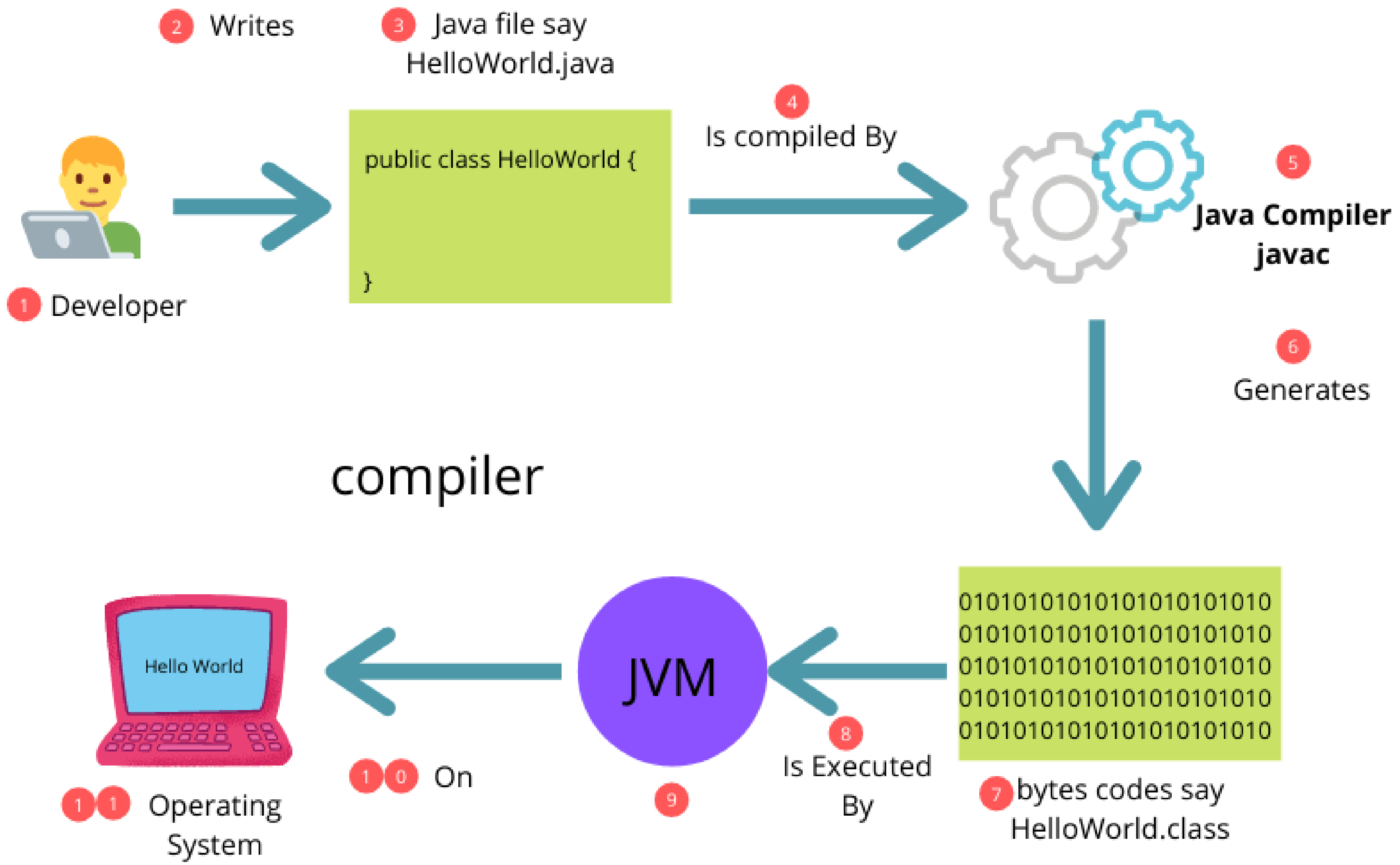
There are four platforms of the Java programming language:

- Java Platform, Standard Edition (Java SE)
- Java Platform, Enterprise Edition (Java EE)
- Java Platform, Micro Edition (Java ME)
- JavaFX

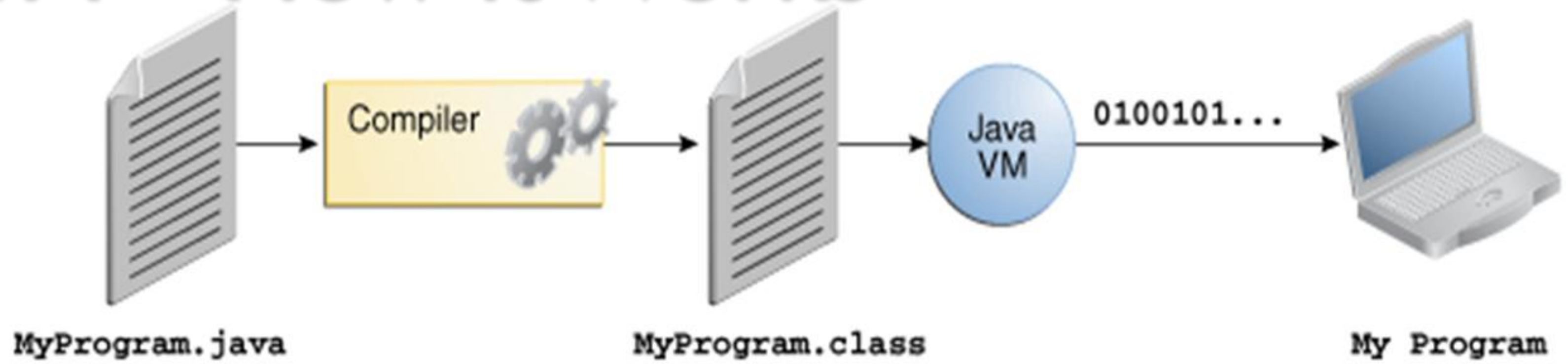
Features of JAVA

- Simple
- Object oriented
- Distributed
- Multithreaded
- Dynamic
- Architecture neutral
- Portable
- High performance
- Robust
- Secure
- And many more…

JAVA – How It Works

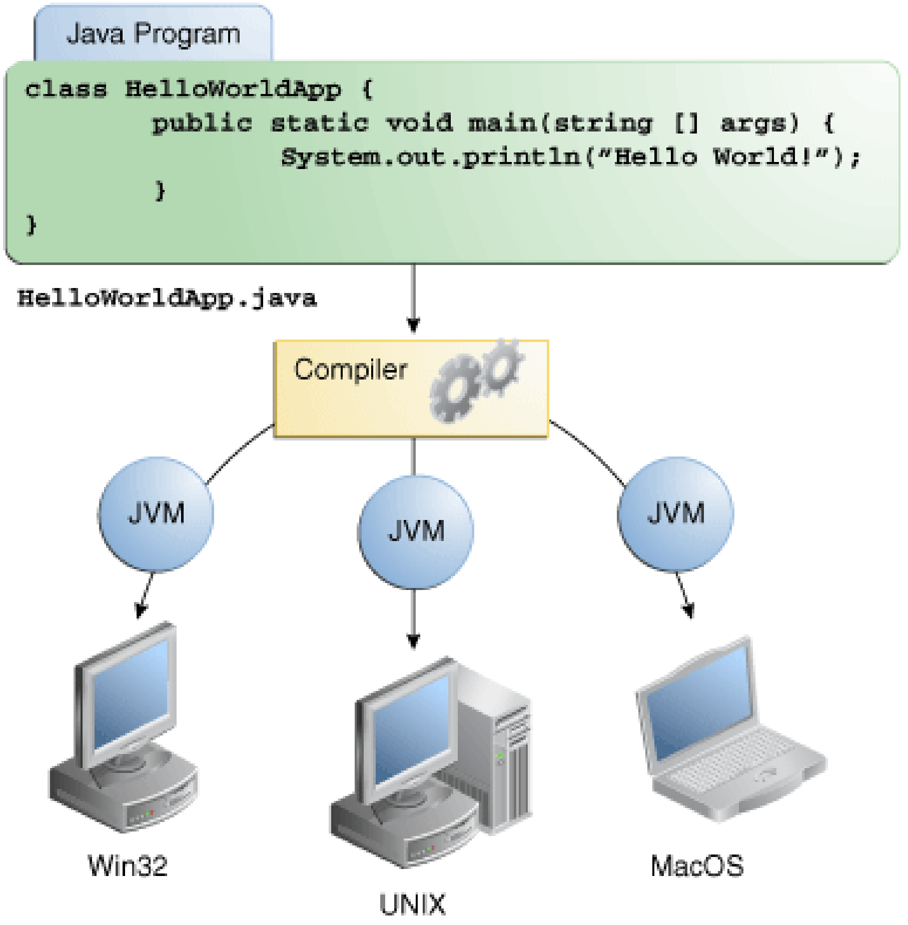


JAVA – How It Works



- In the Java programming language, all source code is first written in plain text files ending with the .java extension.
- Those source files are then compiled into .class files by the javac compiler.
- A .class file does not contain code that is native to your processor; it instead contains bytecodes — the machine language of the Java Virtual Machine1 (Java VM).
- The java launcher tool then runs your application with an instance of the Java Virtual Machine.

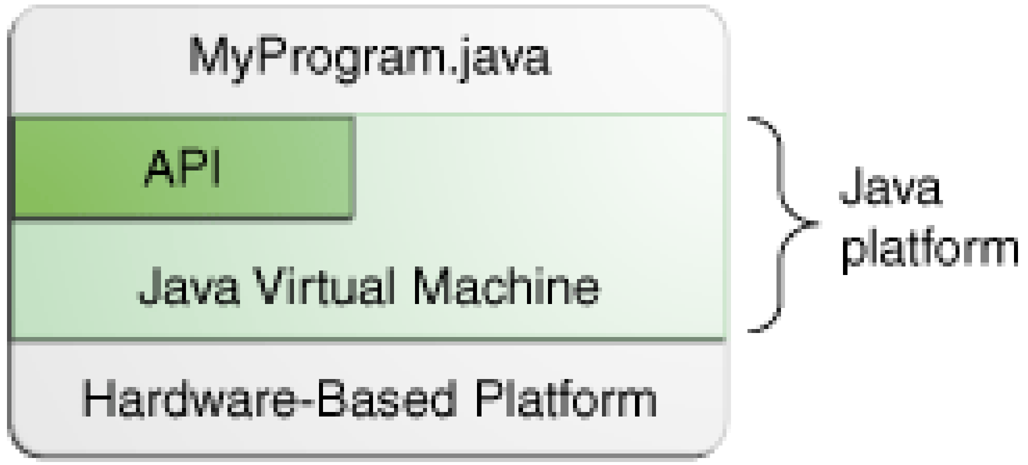
JAVA – How It Works



```
class HelloWorldApp {
        public static void main(string [] args) {
                System.out.println("Hello World!");
        }
}
```

- Through the Java VM, the same application is capable of running on multiple platforms.

JAVA - Platform



- The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.
- The Java platform has two components:
  - The *Java Virtual Machine*
  - The *Java Application Programming Interface (API)*

JAVA -Platform

- The Java Virtual Machine
  - It's the base for the Java platform and is ported onto various hardware-based platforms
- The Java Application Programming Interface (API)
  - The API is a large collection of ready-made software components that provide many useful capabilities.
  - It is grouped into libraries of related classes and interfaces; these libraries are known as packages.

JAVA – Installation

- Go to
  - https://www.oracle.com/java/
- Edit "Environment" variables
- Check the installation
  - Open a Terminal(Windows PowerShell)

    java –version

    javac -version

- Guide
  - https://docs.oracle.com/en/java/javase/19/install/overview-jdk-installation.html#GUID-8677A77F-231A-40F7-98B9-1FD0B48C346A

Writing Your First Program

- Create a new folder with your TG Number
  - TGXXX
- Create a Text Document
- Type the code
- Save as a "<name>.java file

Hello World…!!! ☺

```
public class HelloWorld
{
	public static void main(String[] args)
	{
		System.out.println(“Hello World…!!!);
	}
}
```

Compile Your Program

- Open a Terminal(Windows PowerShell)
- Navigate to the folder
- Compile
  - javac <name>.java

Ex : javac HelloWorld.java

Homework

- Check for the available Java compiler options

Run Your Program

- Run the previously compiled code
  - java <name>.java

Ex : java HelloWorld.java

Homework

- Check for common “Java Command” Options

JAVA – SE 23 – API Documentation

- https://docs.oracle.com/en/java/javase/23/docs/api/index.html

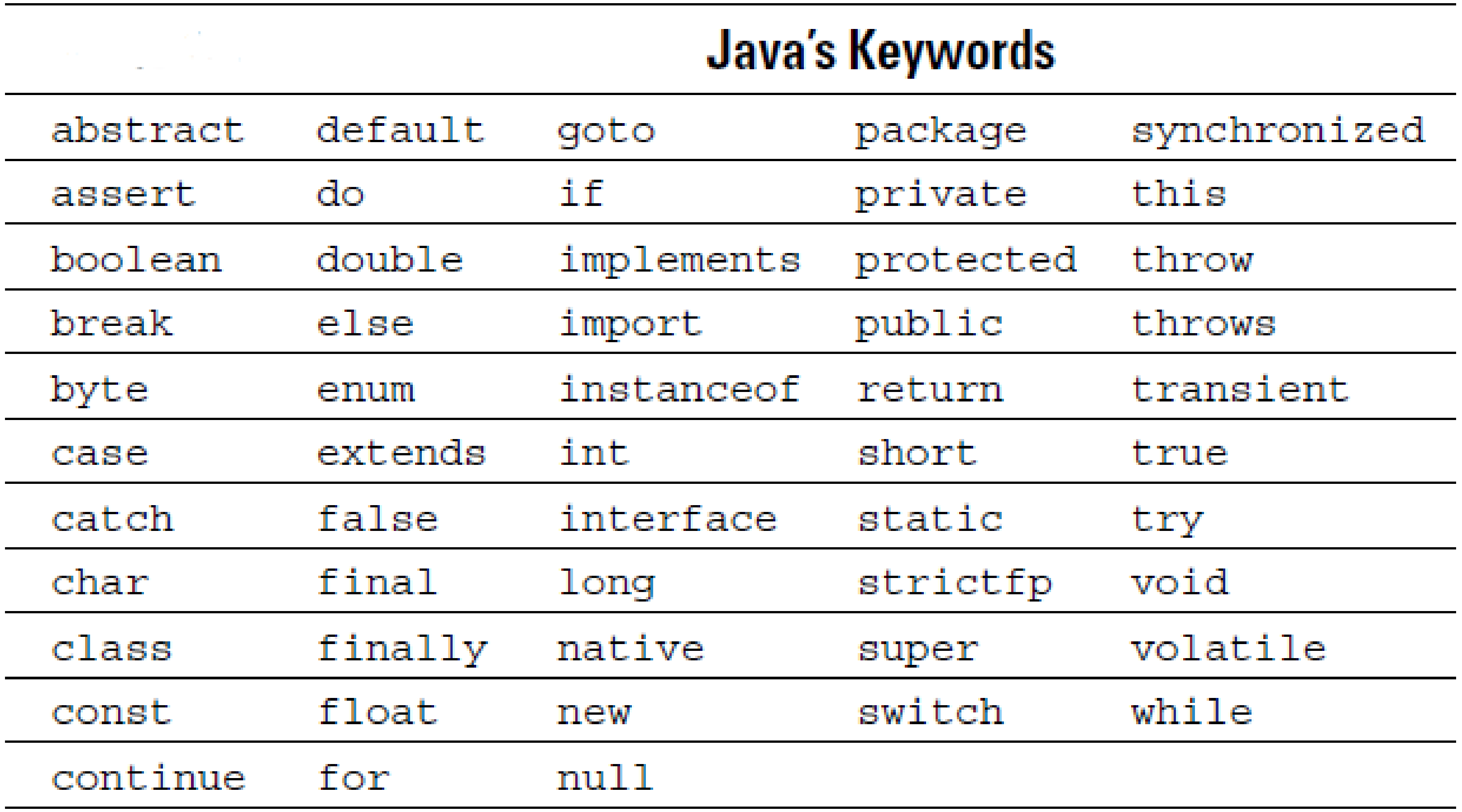
Deep look in to “Hello World”

```
public class HelloWorld
{
        public static void main(String[] args)
        {
                System.out.println(“Hello World…!!!);
        }
}
```

It doesn't work...... why?
It works.......
why?

JAVA - Keywords

- A keyword is a word that has a special meaning defined by the Java programming language.



Java's Keywords

abstract	default	goto	package	synchronized
assert	do	if	private	this
boolean	double	implements	protected	throw
break	else	import	public	throws
byte	enum	instanceof	return	transient
case	extends	int	short	true
catch	false	interface	static	try
char	final	long	strictfp	void
class	finally	native	super	volatile
const	float	new	switch	while
continue	for	null		

JAVA - Statements

- Like most programming languages, Java uses statements to build programs.
- Types of statements
  - Declaration statements
    - simply create variables that you can use to store data.
      - int i;
      - String s = "This is a string";
      - Customer c = new Customer();
  - Expression statements
    - performs calculations etc.
      - i = a + b;
      - salesTax = invoiceTotal * taxRate;
      - System.out.println("Hello,World!");
- There are many more types, Let’s learn on the go

JAVA Class Naming Convention

- Class names should be nouns, in mixed case with the first letter of each internal word capitalized.
- Try to keep your class names simple and descriptive.
- Use whole words-avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as URL or HTML).
  - *class Raster;*
  - *class ImageSprite;*

Class Definition

```
AccessSpecifier class ClassName {

   Description of the variables.
   Description of the constructors.
   Description of the methods.

}
```

The main() method

- Start the program by executing the main method of the class.
- Main method signature,

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

        }
}
```

Obvious similarities to C

- Java syntax has many similarities to C.
- All variables must be declared
- Syntax of expressions and control structures almost identical to C
- C style comments allowed

Exercise 01

- Write a simple java program to print your name and age.
- Extend the above program to print your age in 5 years.

JAVA – White Spaces

- white space refers to one or more consecutive space characters, tab characters, or line breaks.
- All white space is considered the same.

```
x = (y + 5) / z;
```

is identical to this statement:

```
x =
(y + 5) / z;
```

- But with Keywords ???

- *Line breaks:* Place each statement on a separate line.
- *Tabs:* Line up elements that belong together.

JAVA - Blocks

- A block is a group of one or more statements that’s enclosed in braces.
- A block begins with an opening brace ({) and ends with a closing brace (}).
- Between the opening and closing braces, you can code one or more statements.

```
{
        int i, j;
        i = 100;
        j = 200;
}
```

JAVA - Identifiers

- An identifier is a word that you make up to refer to a Java programming element by name.
- Although you can assign identifiers to many types of Java elements, they're most commonly used for the following elements:
    - Classes, such as the HelloWorld class
    - Methods, such as the main method
    - Variables and fields, which hold data used by your program
    - Parameters, which pass data values to methods

JAVA - Identifiers

- Identifiers are case-sensitive. As a result, SalesTax and salesTax are distinct identifiers.
- Identifiers can be made up of upper- or lowercase letters, numerals, underscore characters (_), and dollar signs ($). Thus, identifier names such as Port1, SalesTax$, and Total_Sales.
- All identifiers must begin with a letter. Thus, a15 is a valid identifier, but 13Unlucky isn't (because it begins with a numeral).
- An identifier can't be the same as any of the Java keywords listed in above. Thus, you can't create a variable named *for* or a class named *public.*
- The Java language specification recommends that you avoid using dollar signs in names you create, because code generators use dollar signs to create identifiers. Thus, avoiding dollar signs helps you avoid creating names that conflict with generated names.

JAVA - Comments

- End-of-line comment
  - An end-of-line comment begins with the sequence // and ends at the end of the line.

```
total = total * discountPercent; // calculate the discounted total
// calculate the discounted total
total = total * discountPercent;
```

- Traditional comments
  - A traditional comment begins with the sequence /* , ends with the sequence */, and can span multiple lines.

```
/* HelloWorld sample program.
 This program demonstrates the basic structure
 that all Java programs must follow. */
```

JAVA – Doc Comments

- Search about Java document comments and their usage

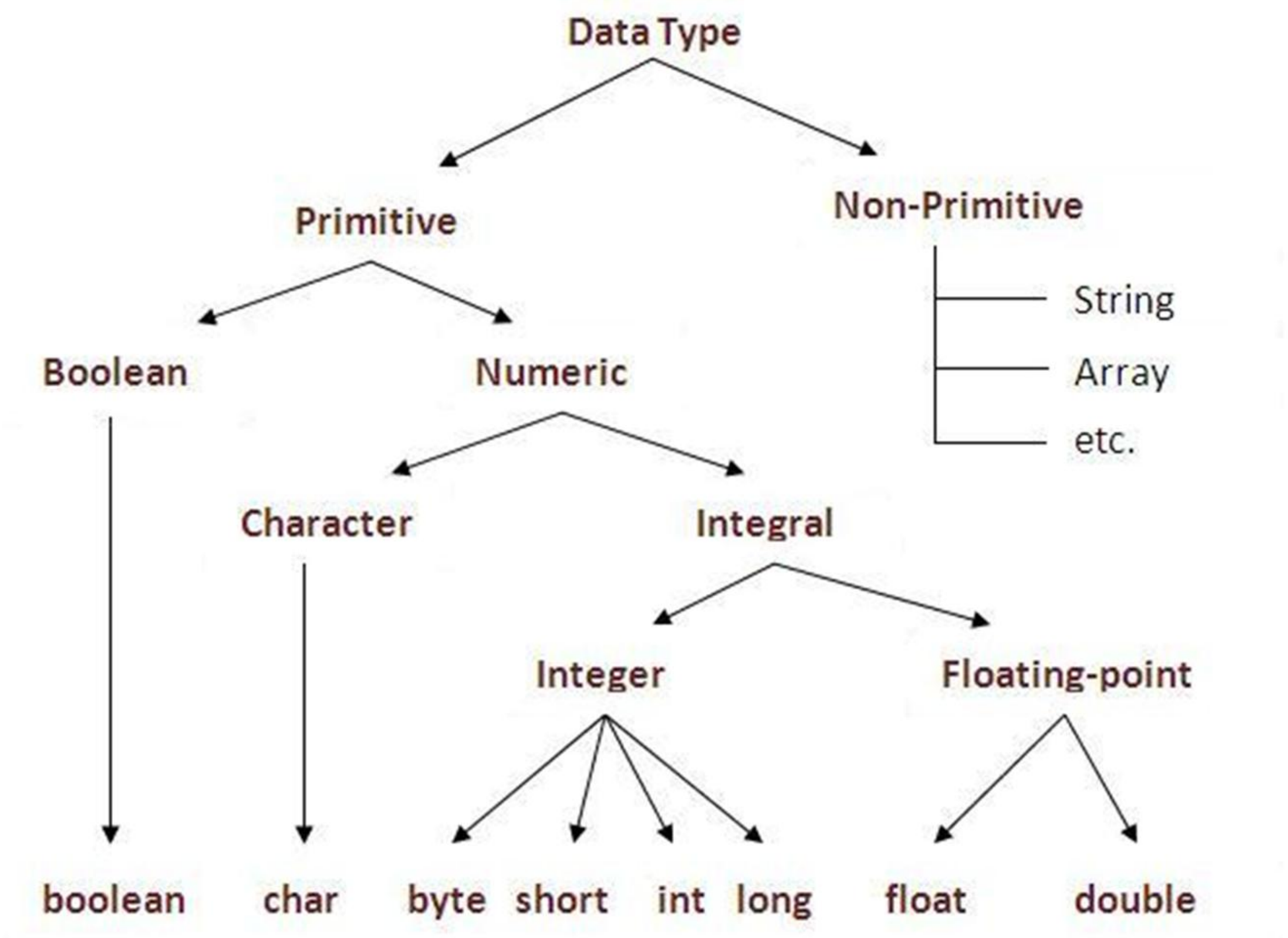
Exercise 02

- Write a simple program to add two numbers.
- Extend the above program to get the difference between the same two numbers, to multiply the same numbers and to divide the first number by the second number.
- Your program should display the result of each operation.

Java Data Types

- Data type specifies the size and the type of values that can be stored in an identifier.
- Each data type uses a Java keyword to be characterized.
- Based on the data type of a variable, the operating system allocates memory and decides what can be stored in the reserved memory.
- Therefore, by assigning different data types to variables, you can store integers, decimals, or characters in these variables.

Java Data Types



JAVA - Variables

- Three types of variables in Java
  - **Local variable**
    - A variable which is declared inside the method is called local variable.
  - **Instance variable**
    - A variable which is declared inside the class but outside the method, is called instance variable . It is not declared as static because then it becomes static variable
  - **Static variable/Class variable**
    - A variable that is declared as static is called static variable. It cannot be local

JAVA Variable Naming Convention

- Except for variables, all instance, class, and class constants are in mixed case with a lowercase first letter.
- Internal words start with capital letters. Variable names **should not start with underscore _ or dollar sign $ characters, even though both are allowed**.
- Variable names should be short yet meaningful.
- The choice of a variable name should be mnemonic- that is, designed to indicate to the casual observer the intent of its use.
- One-character variable names should be avoided except for temporary "throwaway" variables.
  - Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters.
  - int i;   char c;   float myWidth;

Variables Examples

```
public class A{

        static int m=100;            //static variable

        int data=50;          //instance variable

        void method(){
                int n=90;     //local variable
        }  //end method

}//end class
```

Exercise 02 – Redo with variables

- Write a simple program to add two numbers.

- Extend the above program to get the difference between the same two numbers, to multiply the same numbers and to divide the first number by the second number.

- Your program should display the result of each operation.

Exercise 03

- Write the following program in a text document and save it with the name PrePostDemo.java. Compile the program and explain why the value "6" is printed twice in a row when it is executed:

```
class PrePostDemo {
    public static void main(String[] args){
      int i = 3;
       i++;
       System.out.println(i);
       ++i;
       System.out.println(i);
       System.out.println(++i);
       System.out.println(i++);
       System.out.println(i);
    }
}
```

Exercise 04

- Write a method to print each of the following patterns WITHOUT using nested loops in a class called PrintPatterns.

```
#
#
# #
# #
# # #
# #
# # # #
# #
# # # # #
# #
# # # # # #
# #
# # # # # # #
# #
# # # # # # # #
```

```
# # # # # # # #
# # # # # # #
# # # # # #
# # # # #
# # # #
# # #
# #
#
```

```
# # # # # # # #
  # # # # # # #
    # # # # # #
      # # # # #
        # # # #
          # # #
            # #
              #
```

```
          #
        # #
      # # #
    # # # #
  # # # # #
# # # # # #
```

```
      #
     # #
    # # #
   # # # #
  # # # # #
 # # # # # #
# # # # # # #
```

```
   J    a    v     v  a
   J   a a    v   v  a a
J  J  aaaaa    V V  aaaaa
 JJ  a     a    V  a     a
```

```
 +"""""+
[| o o |]
 |  ^  |
 | '-' |
 +-----+
```

Getting user input from keyboard

- There are various ways to read input from keyboard in JAVA.
- **java.util.Scanner** class is one of them.
- The Java Scanner class breaks the input into tokens using a delimiter that is whitespace bydefault.
- It provides many methods to read and parse various primitive values.
- Java Scanner class is widely used to parse text for string and primitive types using regular expression.

Commonly used methods of Scanner class

Method	Description
public String next()	it returns the next token from the scanner.
public String nextLine()	it moves the scanner position to the next line and returns the value as a string.
public byte nextByte()	it scans the next token as a byte.
public short nextShort()	it scans the next token as a short value.
public int nextInt()	it scans the next token as an int value.
public long nextLong()	it scans the next token as a long value.
public float nextFloat()	it scans the next token as a float value.
public double nextDouble()	it scans the next token as a double value.

Example

```
import java.util.Scanner;
public class UserInput
{
        Scanner input = new Scanner(System.in);

        System.out.print(“Enter an Integer value : “);

        int i = input.nextInt();

        System.out.print(“You have entered: “ + i);
}
```

Exercise 05

- Write a program that asks the user's name, and then greets the user by name.

Ex :

```
Please enter your name :  Nuwan
"Hello Nuwan, nice to meet you…!"
```

Summary

- Programming and OOP
- JAVA – History
- JAVA – Features
- JAVA – How It Works
- JAVA - Platform
- JAVA – Installation
- JAVA – Writing Your First Program
- JAVA – Keywords
- JAVA – Statements
- JAVA – White Spaces
- JAVA – Blocks
- JAVA – Identifiers
- JAVA - Comments
- JAVA – Data Types
- JAVA – Variables
- JAVA – User Input
- Exercises

References

- https://docs.oracle.com/javase/tutorial/getStarted/intro/index.html

- How To Program (Early Objects)
  - By H .Deitel and P. Deitel
- Headfirst Java
  - By Kathy Sierra and Bert Bates

Questions ???

Thank You