Semester: fall (2021-2022)

Computer Education Department

Second Grade

Computer Programming I

Introduction to Java applications

Lecture: 1

By: Ms. Slvar A. Arif

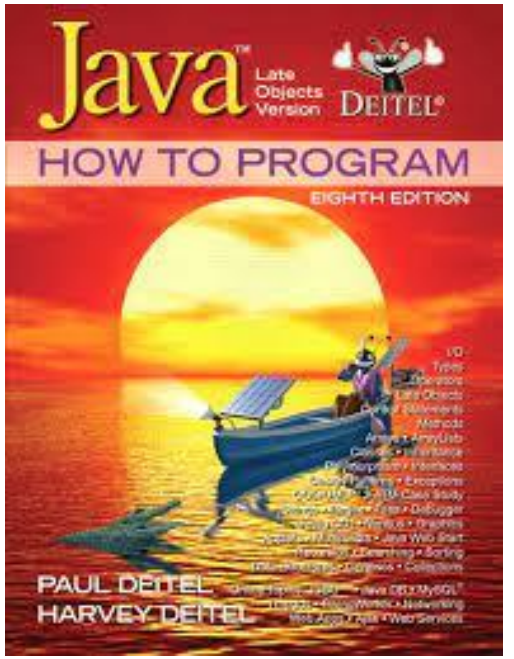
19.10.2021 Tuesday

Textbooks

- Main Textbook:

  Java How To Program (Late Objects) 8th Edition

  by **Paul Deitel and Harvey Deitel**

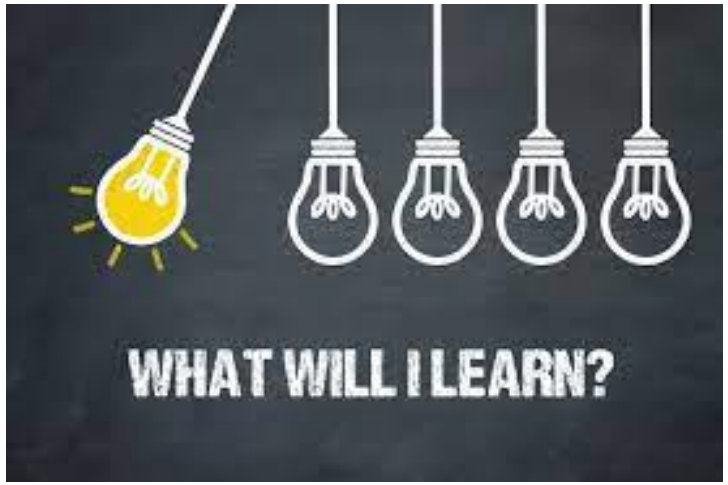


Other Textboxes

- **Thinking in Java 4th Edition** by **Bruce Eckel**
- **Java: A Beginner's Guide, Eighth Edition 8th Edition** by **Herbert Schildt**
- **Sams Teach Yourself Java in 21 Days: Covering Java 7 and Android Original Edition** by **Rogers Cadenhead**

Objectives

- To develop a basic understanding of programming concepts and using these programming concepts in Java language.
- To Introduce programming with Java, variables, basic input/output operations, decision, the loop structures, and basic file operations are covered.
- Learn how to create computer applications using java language.



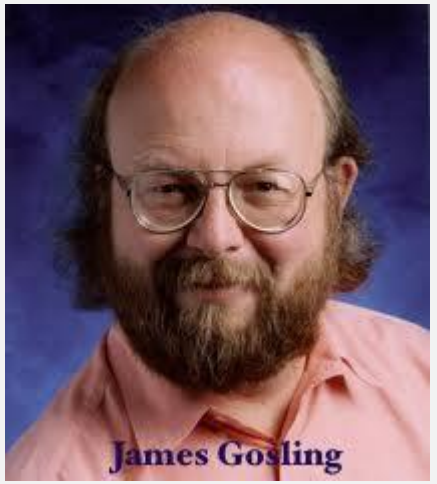
Outline

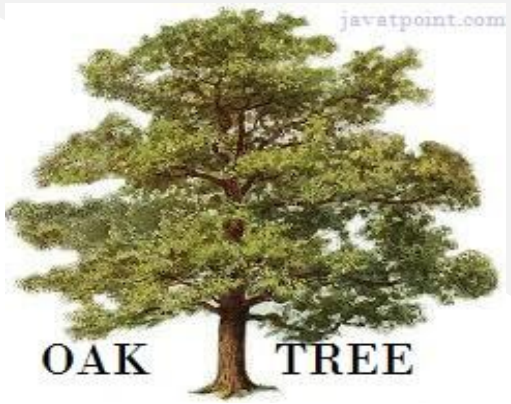
- History of Java
- Java Editions
- Features of Java
- What is Java?
- Java Environment
- Java Program Execution
- Introduction to Java Programming
- First program in Java

History of Java

- **James Gosling -Sun Microsystems**
- **Originally named Oak–then renamed to Java, 1995**
- **JDK Evolutions (Java Development Kit)**

➢ JDK 1.0 (January 23, 1996)
➢ JDK 1.1 (February 19, 1997)
➢ J2SE 1.2 (December 8, 1998)
➢ J2SE 1.3 (May 8, 2000)
➢ J2SE 1.4 (February 6, 2002)
➢ J2SE 5.0 (September 30, 2004)
➢ Java SE 6 (December 11, 2006)
➢ Java SE 7 (July 28, 2011)
➢ Java SE 8 (March 18, 2014)
➢ Java SE 9 (September 21, 2017)





Java Editions

- **J2SE(Java 2 Standard Edition)**

To develop **client-side** standalone applications or applets.

- **J2ME(Java 2 Micro Edition)**

To develop applications for **mobile devices** such as cell Phones.

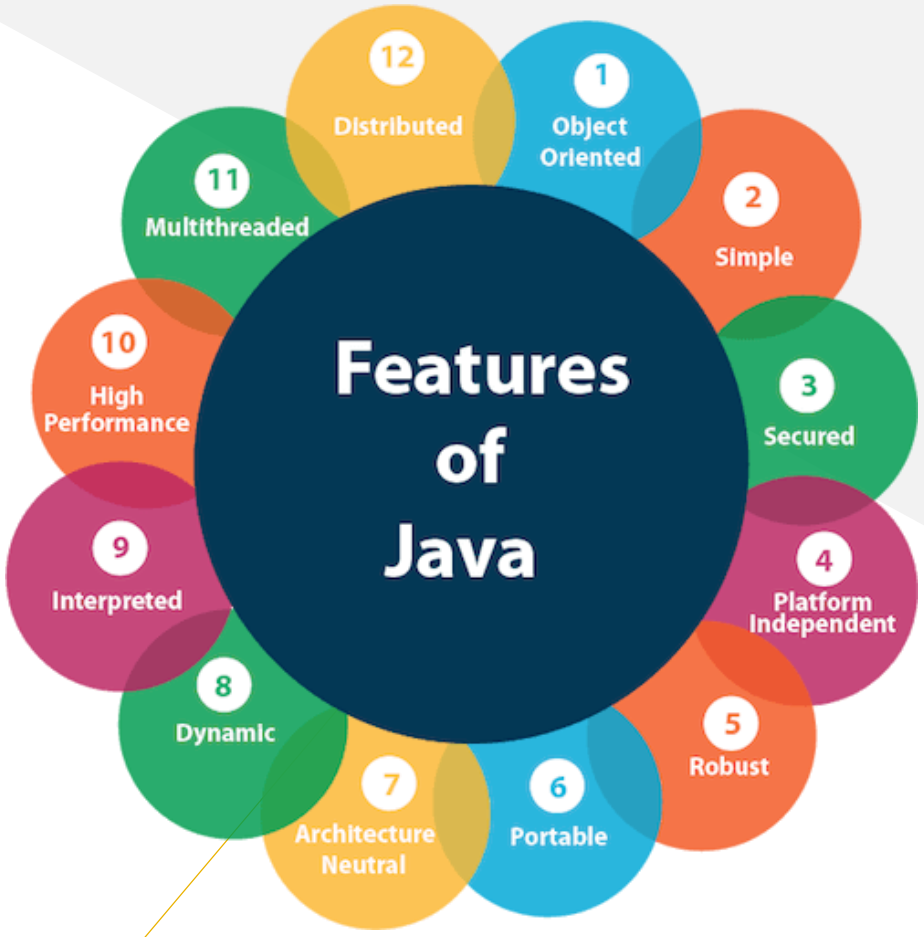
- **J2EE(Java 2 Enterprise Edition)**

To develop **server-side** applications such as Java servlets and Java Server Pages.

Features of Java

➢The most important features of the Java language is given below.

1. Simple
2. Object-Oriented
3. Portable
4. Platform independent
5. Secured
6. Robust
7. Architecture neutral
8. Interpreted
9. High Performance
10. Multithreaded
11. Distributed
12. Dynamic



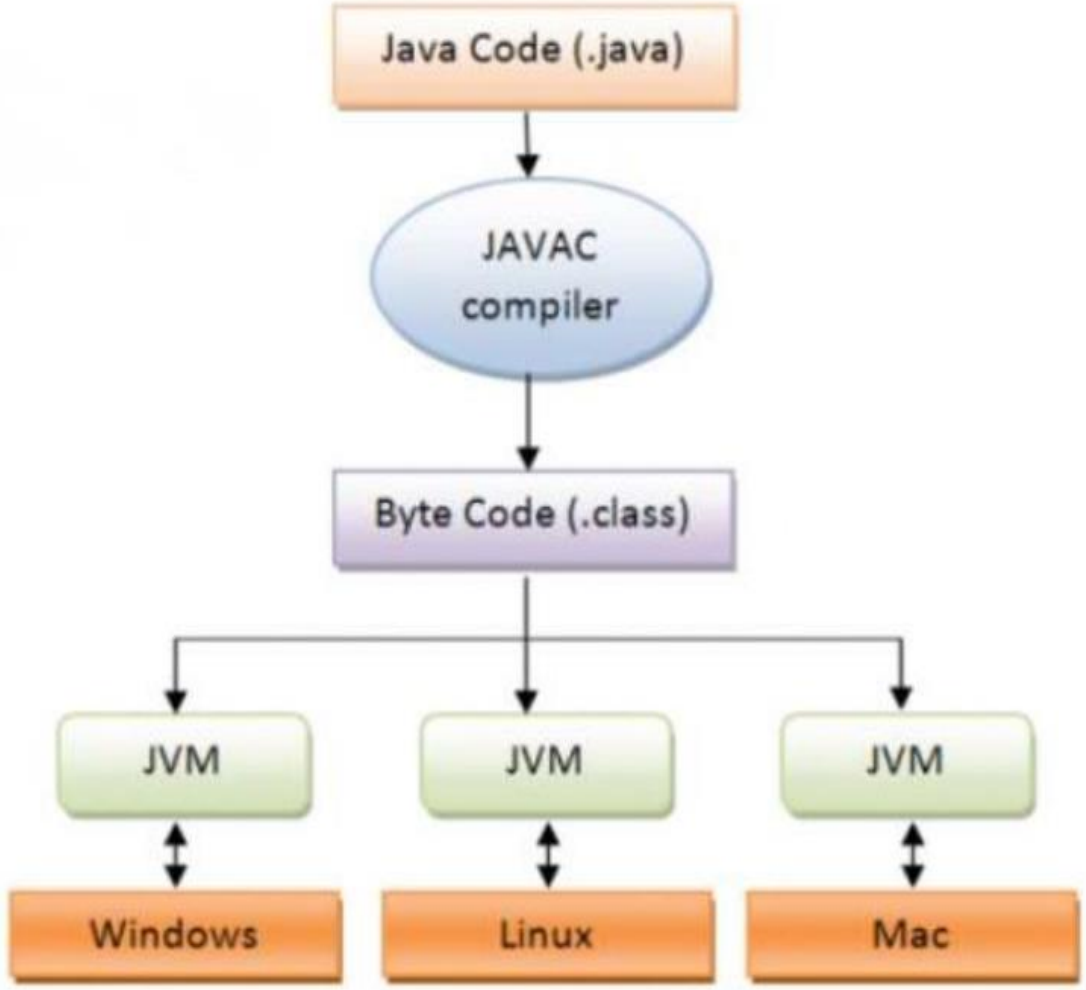
What is Java ?

- **A general-purpose Object-Oriented Language.**
- **Java is platform-independent, Write Once Run Anywhere (WORA).**
- Java is mainly used for application programming. It is widely used in **Windows**-based, **web**-based, enterprise, and **mobile** applications.
- **Widespread acceptance**
- **3 Billion Devices run on Java**
- Java uses both compiler and interpreter.
- Java **source code** is converted into **bytecode** at compilation time.
- The **interpreter** executes this **bytecode** at runtime and produces **output**.
- Java is interpreted that is why it is platform-independent.

Java Environment

- Java includes many development tools, classes and methods
  - Development tools are part of Java Development Kit (JDK) and
  - The classes and methods are part of **Java Standard Library** (JSL), also known as **Application Programming Interface** (**API**).
- JDK constitutes of tools like java compiler, java interpreter and many.
- API includes hundreds of classes and methods grouped into several packages according to their functionality.

Java Program Execution



Introduction to Java Programming

Java application programming

- Display messages.
- Obtain information from the user.
- Arithmetic calculations.
- Decision-making fundamentals.
- **Application**
  - Executes when you use the java command to launch the Java Virtual Machine (JVM).
- **Sample program**
  - Displays a line of text.
  - Illustrates several important Java language features.

First program in Java

Printing a Line of Text

```
1  // Fig. 2.1: Welcome1.java
2  // Text-printing program.
3
4  public class Welcome1
5  {
6     // main method begins execution of Java application
7     public static void main( String args[] )
8     {
9        System.out.println( "Welcome to Java Programming!" );
10
11    } // end method main
12
13 } // end clazss Welcome1
```

```
Welcome to Java Programming!
```

First program in Java (Cont.)

Printing a Line of Text

```
1  // Fig. 2.1: Welcome1.java
```

- Comments start with: //
  - Comments ignored during program execution
  - Document and describe code
  - Provides code readability
- Traditional comments: /* ... */

```
/* This is a traditional
   comment. It can be split over many lines */
```

-

```
2  // Text-printing program.
```

- Note: line numbers not part of program, added for reference

Note: Every program should begin with a **comment** that explains the purpose of the program.

First program in Java (Cont.)

Printing a Line of Text

```
3
```

- Blank line
  - Makes program more readable
  - Blank lines, spaces, and tabs are white-space characters
    - Ignored by compiler

```
4  public class Welcome1
```

- Begins class declaration for class `Welcome1`
  - Every Java program has at least one user-defined class
  - Keyword: words reserved for use by Java
    - `class` keyword followed by class name
  - Naming classes: capitalize every word
    - `SampleClassName`

Note: Use blank lines and space characters to enhance program readability..

First program in Java (Cont.)

Printing a Line of Text

```
4  public class Welcome1
```

- Java identifier
  - Series of characters consisting of letters, digits, underscores (_) and dollar signs ($)
  - Does not begin with a digit, has no spaces
  - Examples: `Welcome1`, `$value`, `_value`, `button7`
    - `7button` is invalid
  - Java is case sensitive (capitalization matters)
    - `a1` and `A1` are different

Note: By convention, always begin a class name's identifier with a **capital letter** and start each subsequent word in the identifier with a capital letter.

First program in Java (Cont.)

Printing a Line of Text

```
4  public class Welcome1
```

- Saving files
  - File name must be class name with `.java` extension
  - `Welcome1.java`

```
5  {
```

- Left brace {
  - Begins body of every class
  - Right brace ends declarations (line 13)

Note:

- It is an **error** for a public class to have a file name that is not identical to the class name (plus the .java extension) in terms of both spelling and capitalization.
- It is a **syntax error** if braces do not occur in matching pairs.
- It is an **error** not to end a file name with the .java extension for a file containing a class declaration.

First program in Java (Cont.)

Printing a Line of Text

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

- Part of every Java application
  - Applications begin executing at `main`
    - Parentheses indicate `main` is a method
    - Java applications contain one or more methods
  - Exactly one method must be called `main`
- Methods can perform tasks and return information
  - `void` means `main` returns no information

First program in Java (Cont.)

Printing a Line of Text

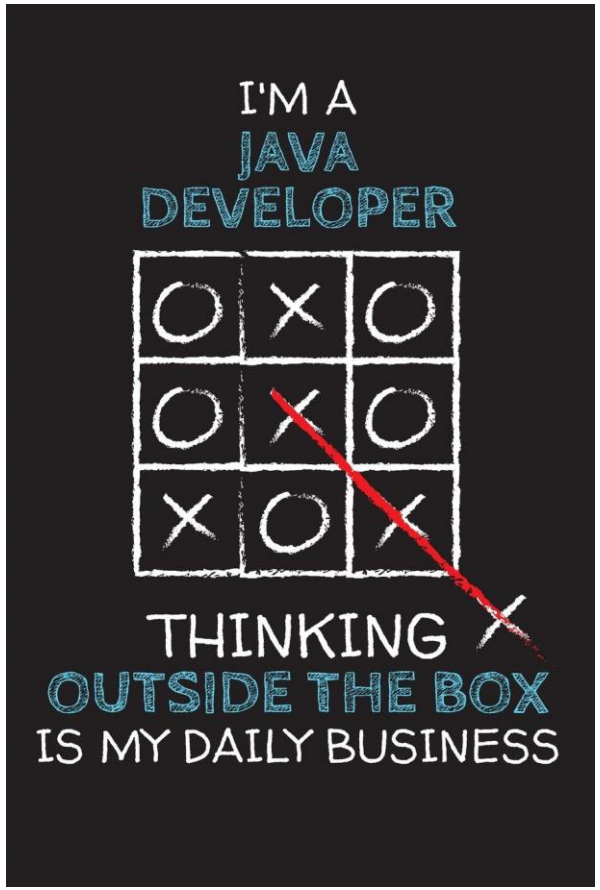
```
9         System.out.println( "Welcome to Java Programming!" );
```

- Instructs computer to perform an action
  - Prints string of characters
    - String – series of characters inside double quotes
  - White-spaces in strings are not ignored by compiler
- `System.out`
  - Standard output object
- Method `System.out.println`
  - Displays line of text
- This line known as a statement
  - Statements must end with semicolon ;

Note: Omitting the semicolon ; at the end of a statement is a syntax error.

Homework

Write a java program which displays your full name to the user.



Next Lecture

1. Modifying our first Java program .
2. Displaying a Text with Printf.
3. Adding Integers Java program.