



Republic of the Philippines
Department of Education
REGION III
SCHOOLS DIVISION OFFICE OF NUEVA ECija

LEARNING ACTIVITY SHEET
SPECIAL PROGRAM IN ICT 8
BASIC PROGRAMMING
Second Quarter, Week 1

Features of Java

BACKGROUND INFORMATION FOR LEARNERS

What is Java Programming Language?

Java is a widely used programming language and is designed for the distributed environment of internet. It is a general-purpose programming language that is concurrent, class-based, and object-oriented. It is free to access and we can run it on all the stages. Java follows the principle of WORA (Write Once, Run Anywhere), and is platform-independent. It is also simple and easy to learn. If you want to print say (“Hello, World!”), you would type:

Java Hello World Example-

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

HISTORY OF JAVA

Java to begin with showed up in 1995 as Oak. This was a reference to the Oak tree that stood a far distance Gosling’s office. Sometime recently it seem discover its popularity as Java (finding its roots in coffee, which in turn is credited to Java- an island in Indonesia), they too chosen to call the extend Green.

JAMES GOSLING, founder of Java Version 1.0 rolled out in 1996 when Sun Microsystems promised the principle of WORA (Write Once, Run Anywhere).

Then came along Java 2 (J2SE 1.2) in December 1998-1999. J2EE was for enterprise applications.

Then in 2006, boosting its marketing capabilities, Sun renamed new J2 versions as Java EE, Java ME, and Java SE.

September of 2018 marked the release of Java SE 11 (LTS).



March of 2019 marked the release of Java SE 12 (LTS).

By September 10th 2019, Java SE 13 will get to see the light of day.

Today from web applications and desktop GUI's to the Internet of Things and self-driving cars, Java is everywhere.

FEATURES OF JAVA PROGRAMMING

Speaking of its popularity let's find out what makes it worth. In this tutorial, we will discuss a list of powerful Features of Java :

1. Simple

While Java is verbose, its syntax is similar to C++. This makes it easier to transition from C++ to Java. It has removed rarely-used features like operator overloading and explicit pointers.

2. Object-oriented

Java mainly focuses on objects rather than processes. Java follows the principles of Object-Oriented Programming (OOP):

- Objects
- Classes
- Inheritance
- Encapsulation / Data hiding
- Abstraction
- Polymorphism

Note: Java isn't a purely object-oriented language as it allows primitive data types.

3. Platform-independent

Platform independent implies that the java source code can run on numerous operating systems. Java code ought to run on any machine that doesn't require any extraordinary computer program to be introduced, however the JVM has to be display within the machine. Java code is compiled into bytecode, which is platform-independent. You'll be able run it on Windows, Linux, Mac OS, etc.

4. Portable

Java is portable because Java code can be executed on all the major platforms. Once you've compiled your Java source code to bytecode, those files can be used on any Java-supported platform without modification, unlike other languages, which requires compiling the machine code for each platform, for example, a separate .exe for 32-bit vs 64-bit environments. Java is portable because you can carry bytecode over to any other platform it runs on.

5. Robust

The following features make Java strong and powerful:

- There is no use of explicit pointers
- Strong memory management

- Automatic garbage collection is done so you don't need to delete the unreferenced objects manually.
- Exception handling and type-checking mechanisms

6. Secure

- As we said in the previous section, the lack of pointers makes Java powerful and secure.
- It is also true that Java programs run inside a virtual machine sandbox.
- The JRE (Java Runtime Environment) has a Classloader to dynamically load classes into the JVM (Java Virtual Machine). It separates the package for the classes of the local file system imported from networks, this adds to the security.
- The Bytecode Verifier inspects fragments of code for illegal code that can bypass access.
- The Security Manager decides what resources to allot to a class. Such access can include reading and writing files.
- Java can help you develop virus-free systems.

7. Architecture Neutral

Primitive languages were not neutral to the architecture of development environments. For example, C had different sizes of datatypes for 32 bit systems and 64 bit systems. So did a number of other languages too.

Java decided to be neutral to all platforms which dramatically increased collaboration efficiency. Java code does not compile its code to platform-specific byte code but it's compiled into platform independent bytecode. This means that the generated class file can now run on different machines running different environments, different operating systems with the only requirement of a Java Virtual Machine to be present in each machine. This makes Java an architecturally robust and flexible programming language.

8. High Performance

Java's bytecode makes it a lot faster than other languages because of its innate similarity to native code. Native code is the code which is processor specific, ie. it has to be compiled to run with a specific processor, like Intel's x86 class processor.

Microsoft's Intermediate language and Java Bytecode has native code for execution of an in-time compiler for faster performance. However, Java is an interpreted language which means that it's slower than compiled languages like C and C++.

Java is faster than a lot of other languages like python, however, this is an abstract concept as python also takes less time to develop due to the syntax and easily fabricated code design

9. Multithreaded Programming

Java supports multithreaded programming i.e, it supports multiple operations running at the same time. We can think of a thread as an individual operation or parts of the program using the processor. It increases the performance by decreasing the development time needed for a particular software. The coding for a particular software becomes streamlined. The maintenance cost drops. However, all of these processes share the same memory slots as individual processes use the memory efficiently. As a multiprocessor can effectively execute multiple threads at the same time, multiprogramming is a boon to the developer community. Java manages to cover all these points which makes it a super-efficient language.

10. Distributed Language

Java isn't distributed as such when it's compared to the exact definition of a distributed system. It can be used for transfer and execution of programs by remote computers on the local machines from the internet. It supports third party libraries to create and support web services. It's a rule based language model. Java loads/invokes the objects or functions only when the program needs it. Java finalizes invoking instructions during runtime. Ex- Runtime Polymorphism i.e function overriding.

RMI and EJB are some of the applications which find use in distributed applications. RMI stands for Remote Method Invocation. As the name suggests it can invoke other methods running in other JVM's. It uses a stub and a skeleton for communication. However EJB stands for Enterprise Java Bean. It helps in developing scalable secure and robust applications.

11. Dynamic Nature

It's a requirement based language model. Classes are not loaded all at once. They jump into action only when an invoke operation executes or some data about the class is needed in the memory. Java finalizes invoking instructions during runtime. Ex- Runtime Polymorphism i.e function overriding.

JAVA ARCHITECTURE – The Java Environment

In this section of the Java tutorial, we will see the introduction of JVM, JDK, JRE.

1. JVM (Java Virtual Machine)

The main purpose of Java Virtual Machine is to provide a runtime environment in which bytecode executes. It is platform-dependent, and has the following tasks:

- Load code
- Execute code
- Verify code
- Provide a runtime environment

2. JRE (Java Runtime Environment)

This is a collection of tools that together allow the development of applications and provide a runtime environment. The JVM is a part of JRE. This is like JVM, platform-dependent.

3. JDK (Java Development Kit)

JDK is Kit which provides the environment to develop and execute the Java program. It includes Development Tools to provide an environment to develop your Java programs and JRE to execute your java code. Along with the JRE, JDK contains other resources like the interpreter/loader (java), the compiler (javac), an archiver (jar), and a documentation generator (Javadoc). Together these components help you to build Java programs.

ADVANTAGES AND DISADVANTAGES OF JAVA

Let's discuss what makes Java great, and what doesn't. In this section, we will get to know the advantages and disadvantages of Java.

Advantages of Java

- Java is platform independent because we can run Java code on any machine that doesn't need any special software to be installed, JVM does it.
- Java is object-oriented because of its classes and objects.
- The main reason for Java being secure is pointers, Java doesn't use pointers.
- In Java, we can execute many programs simultaneously, therefore, multithreading can be achieved.
- Java is robust because it has many features like garbage collection, no use of explicit pointers, exception handling.
- Java is a High-level language which makes it easy to understand.
- Efficient memory management is done in Java Programming Language.
- Explore the Advantages and Disadvantages of Java Programming Language

Disadvantages of Java

- Being a high-level language, it must deal with the compilation and abstraction levels of a virtual machine. Java exhibits poor performance the main reason is garbage collector, bad caching configuration, and deadlocks, among processes.
- Java has very few GUI builders – Swing, SWT, JSF, and JavaFX among the more popular ones.
- To write code to carry out a simple set of activities, you could end up writing long, complicated code. This can affect readability but ensures that programmers type in exactly what needs to be done. When we compare Java to Python, we observe Python does not need semi-colons, parentheses, or curly braces, and has a visibly shorter code.

APPLICATIONS OF JAVA PROGRAMMING

Java is widespread, the following are some of the areas in which we find java usable:

1. Desktop applications
2. Web applications
3. Mobile applications (Android)
4. Cloud computing
5. Enterprise applications
6. Scientific applications
7. Operating Systems
8. Embedded systems
9. Real-time software
10. Cryptography
11. Smart cards
12. Computer games
13. Web servers and application servers

JAVA PLATFORM EDITIONS

In this part of the Java tutorial, we will discuss the Java platform edition:

1. Micro Edition – J2ME (Java ME)
2. Standard Edition – J2SE (Java SE)
3. Enterprise Edition – J2EE (Java EE)
4. Java Card

a. Java ME (Micro Edition – J2ME)

This is useful for developing small devices like mobile phones. The Java ME API is a subset of the Java SE API.

b. Java SE (Standard Edition – J2SE)

The Java SE API holds the core functionality of Java – from basic types and objects to high-level classes for GUI, database access, networking, and security.

c. Java EE (Enterprise Edition – J2EE)

This one is built on top of Java SE. It delivers an API and a runtime environment for the development and running of large-scale, secure network apps.

d. Java Card

This edition lets us build for smart cards using Java.*

TOP COMPANIES USING JAVA

When we say Java is an immensely popular language, we aren't simply talking about freshers. The following big names still use Java to build or improve their products and services:



LEARNING COMPETENCY

- Identify the features of Java and the importance of Studying Java.
- Familiarize with the software requirements to set up and run Java.

ACTIVITIES

Activity No. 1. Watch the video using the link below and answer the given question.

<https://www.youtube.com/watch?v=M99r7VS572M>

Intro to Fundamentals of JAVA Programming Tagalog Tutorial

1. Why do we have to learn Java Programming language?

Activity No. 2: Watch the video using the link below and answer the given question.

<https://www.youtube.com/watch?v=9Kk1dGkpGPI>

Software Requirements for Java

1. What are the basic software requirements for Java?

Activity No. 3: Answer the attached Google Form in our Google Classroom

REFERENCES

<https://data-flair.training/blogs/features-of-java/>

<https://www.youtube.com/watch?v=M99r7VS572M>

<https://www.youtube.com/watch?v=9Kk1dGkpGPI>

Compiled by: CHARITO DC. SANTOS

Name of Writer

Noted by: LABERNE A. LADIGNON, JR

Division ICT Coordinator/ OIC EPS