

Bule Hora University

College of Informatics

Department of Information Technology

Object oriented Programming in Java (ITec2052)

Chapter-1:Introduction to Object oriented Programming

➤ Content of chapter-1

- ✓ **Introduction to object oriented Programming**
- ✓ Types of programming paradigm
- ✓ Overview of OO principles
- ✓ Overview of Java Programming and types of Java Program
- ✓ Definition of Java Application, Java Applets Editing, Compiling and Interpreting

Organized by: Negessa W and Sadik A.

March, 2020

Introduction to object oriented Programming (OOP) in java

- OOPs refers to languages that uses objects in programming.
- Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc in programming.
- In OOP, computer programs are designed in such a way where everything is an object that interact with one another.
- An “Object” is the key component of Object-oriented programming. An Object may contain data (fields or variables) or code (methods or procedures). The creation of these objects is based on a **programmer-defined blue-print** also known as a **Class**.

- Classes are like ideas. Objects are concrete manifestations of those ideas.
- An object can be defined as a data field that has unique attributes and behavior. It is **instance** of class.
- Example of class and object in java.

```
public class MyClass {  
    int x = 5;  
    public static void main(String[] args) {  
        MyClass myObj = new MyClass();  
        System.out.println(myObj.x);  
    }  
}
```

OOP.....

- **OOP** is a programming concept that works on the principles of abstraction, encapsulation, inheritance, and polymorphism.
- It allows users to create the objects that they want and then, create methods to handle those objects.
- The basic concept of OOPs is to create objects, re-use them throughout the program, and manipulate these objects to get results.
- OOP is a high-level programming language based on the object-oriented model.
- Java is an example of object-oriented programming language.

What is programming paradigms?

- Is a fundamental style of computer programming used for solving computational problems.
- **Problem** → **logic** → **algorithm** → **programming paradigm**
- Major types of programming paradigms are:
 - ✓ **Imperative**,
 - ✓ **Structural**,
 - ✓ **Procedural** and **Object-Oriented**.

□ **Imperative :**

- it is oldest approach.
- Closest to the actual mechanical behavior of a computer ⇒ original imperative languages were abstractions of assembly language.
- A program is a **list of instructions** that change a memory state until desired end state is achieved.
- Useful for quite simple programs.
- Difficult to scale.

❑ Structured

- Structured program is sequencing, selection, and iteration are sufficient to express any computable function.
- structured: subroutines/functions, block structures, and loops, and tests.

❑ **Procedural**

- Evolution of structured programming.
- Divide the code in procedures: routines, subroutines, modules methods, or functions.
- Advantages
 - ❖ Division of work.
 - ❖ Debugging and testing.
 - ❖ Maintenance.
 - ❖ Reusability.

❑ **Object oriented programming paradigm**

- It is one of most popular programming paradigms.
- Program represents real world objects
 - objects have properties
 - Objects exhibit behavior and interacts with other objects
- Useful for development of enterprise applications
- It has features for data security, code reuse and maintainability
- Example: C++ ,java and python

OO principles

❑ Polymorphism

- It means one name many forms.
- Polymorphism is the capability of a method to do different things based on the object that it is acting upon.
- In other words, polymorphism allows you define one interface and have multiple implementations

❑ Encapsulation

- Encapsulation simply means binding object state(fields) and behavior(methods) together. If you are creating class, you are doing encapsulation.

❑ **inheritance**

- The process by which one class acquires the properties and functionalities of another class is called **inheritance**.
- Inheritance provides the idea of reusability of code and each sub class defines only those features that are unique to it, rest of the features can be inherited from the parent class.

❑ **Abstraction in Java**

- When we hide the unnecessary detail and defining the useful (relevant) detail, then the procedure is said to be an *abstraction*.
- **Main advantage:** the user gets data according to their needs but they don't need to use unnecessary data.

Overview of Java Programming and types of Java Program

- **Java** is an object-oriented language similar to C++, but with advanced and simplified features. Java is **free to access** and can **run on all platforms**.
- **Java is Simple**
 - Because of the stylistic simplicity of the language, many people learn it quickly.
- **Java is Object Oriented**
 - The OOP paradigm has risen in popularity
- **Java is Interpreted**
 - Java source code is passed to a compiler that generates the byte code.
 - JVM interprets the byte code at runtime and executes it.

- **Java is portable**
 - Java applications can run practically anywhere.
 - You can truly write the code once and run it anywhere.
- **Java is Robust**
 - Robust code is reliable code.
 - Java is considered a *strongly typed* language.
 - JVM insures robustness managing all memory automatically.
 - Java also includes an extensible mechanism for exception handling.

- **Java is Secure**

- Because Java technology is so prevalent on networks and in today's enterprise systems, its security aspects are vital.

- **Java is Multithreaded**

- Java provides a feature with which you can develop an application that can do more than one thing at a time.

- **Java is High Performance**

- Java versions have performed at the speed that developers demand.

Types of java program

1. ***Stand-alone applications*** are those java programs that can be developed and executed on a stand-alone local computer.(which we can execute from the command prompt).

The stand-alone application can be executed without the browsers (Internet connectivity).

A java program is not embedded within HTML or any other language and can Stand on its own.

It is referred to as Application program.

2. *Applet Programs* are small java programs developed for internet applications.

Applets are embedded in HTML documents.

Applet programs can be run using the applet viewer or web browser.

For example, when you use java to enhance a WWW page, the java code is embedded within HTML code.

3. Java Servlets

- Java servlets provide web developers with a simple, consistent mechanism for extending the functionality of a web server and provide dynamic behavior for web applications.

Java Applications:

- Mobile Applications
- Desktop GUI Applications
- Web-based Applications
- Enterprise Applications
- Scientific Applications
- Gaming Applications
- Big Data technologies
- Business Applications
- Distributed Applications
- Cloud-based Applications

Common Java application Editor

- **NetBeans** is an open-source Integrated Development Environment written in Java.
- Development of NetBeans has really accelerated since it became part of the Apache project.
- The NetBeans IDE can be used to develop in Java, but also supports other languages. These include PHP, C/C++, and HTML5.
- **Eclipse** is another free Java IDE for developers and programmers. It is mostly written in Java. Eclipse lets you create various cross-platform Java applications for use on mobile, web, desktop and enterprise domains.

Is Java Compiled or Interpreted Programming language?

- Java is both compiled and interpreted program, but it is not completed. What **javac** (**Java compiler**, which comes along JDK) does is pseudo compilation, it doesn't convert Java source code into native code, which can directly be executed by CPU, real compilation into native code is done by another program called Just in Time compiler, also known as **JIT**.
- This is actually an optimization done on JVM by Java platform engineers.

- When JVM interprets Java byte code, it also gathers useful statistics, like which part of code is hot and always run. Once JVM has enough data to make such decision, JIT can compile that part of code e.g. method or block into native code.
- This native code will then directly be executed by machine, without interpreting by JVM.
- JIT provides immense performance boost to Java application and this is one more reason why Java is also used to write high performance application like electronic trading systems, algorithmic gateways etc, along with native languages like C and C++.
- In short, ***Java is both compiled and interpreted language***. It compiles using [javac](#) and JIT, and interprets using Java virtual machine.

Here is the sequence of things normally happens from writing to execution of a Java program :

- Programmer write source code and store that into a .java file
- javac compiles .java file and creates class files, which contains byte code. This byte code is main reason of Java being platform independent.
- JVM executes these class files and gathers statistics of execution run
- After certain threshold, when JVM has enough data to make decision, JIT compiles frequently used byte codes into native code, which is then directly executed by platform. This provides performance boost to Java application

Assignment 5%

1. Discuss JDK, javac and JVM?
2. Discuss OOP principles?
3. Discuss class and object?