Session 1

Programming and Java

Course Description

- Covers the concepts of OOP languages and systems
  - fundamental abstraction, modularity and encapsulation mechanisms in OOP
- Advanced OOP concepts
  - polymorphism and operator overloading; message passing via generic functions; late versus early binding times; and inheritance mechanisms and their relationship to the type systems of programming languages.

Prerequisite

- Basic knowledge of
  - Computer Programming

Course Contents

- Object-Orientation
- Encapsulation
- Abstraction
- Objects and Classes
- Method Overloading
- Operator Overloading [C# or C++]
- Inheritance
- Method overriding
- Composition/Aggregation
- Polymorphism
- Exception Handling
- Object Oriented Modeling and Use Case Diagrams

Course Learning Outcomes (CLO)

CLO #	CLO Statements	Bloom's Taxonomy Code	Associated PLO
CLO 1	Describe the fundamental concepts of Object Oriented Programming for example Constructors, Destructors, Encapsulation, this pointer, Inheritance, Aggregation, Composition etc.	C1	PLO1 - Engineering Knowledge
CLO 2	Explain and compare some of the advanced OOP concepts like subtyping, specialization, Multilevel and Hierarchical inheritance, native, final modifiers, static and dynamic binding etc.	C2	PLO2 - Problem Analysis
CLO 3	Apply OOP programming concepts for problem solving.	C3	PLO3 - Designing and Development
CLO 4	Present an advanced programming topic confidently and professionally.	A2	PLO9 – Individual and Team Work

The Text/Reference books

- Timothy A. Budd, **The Introduction to Object-Oriented Programming**, Addison-Wesley
- Avinash C. Kak, **Programming with Objects: A comparative presentation of object-oriented programming with C++ and Java**, John Wiley & Sons
- Jacobson, Christerson, Jonsson, Overgaard , **Object-Oriented Software Engineering**.
- Bruce Eckel, **Thinking In Java**
- Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson, **Professional C#**, Wrox Press Ltd.

Session Objectives

- To understand the relationship between Java and the World Wide Web.
- To know Java’s advantages .
- To distinguish the terms API, IDE, and JDK .
- To write a simple Java program.
- To create, compile, and run Java programs.
- To understand the Java runtime environment.
- To know the basic syntax of a Java program.

Programs

Computer *programs*, known as *software*, are instructions to the computer.

You tell a computer what to do through programs. Without programs, a computer is an empty machine. Computers do not understand human languages, so you need to use computer languages to communicate with them.

Programs are written using programming languages.

Programming Languages

Machine Language   Assembly Language   High-Level Language

Machine language is a set of primitive instructions built into every computer. The instructions are in the form of binary code, so you have to enter binary codes for various instructions. Program with native machine language is a tedious process. Moreover the programs are highly difficult to read and modify. For example, to add two numbers, you might write an instruction in binary like this:

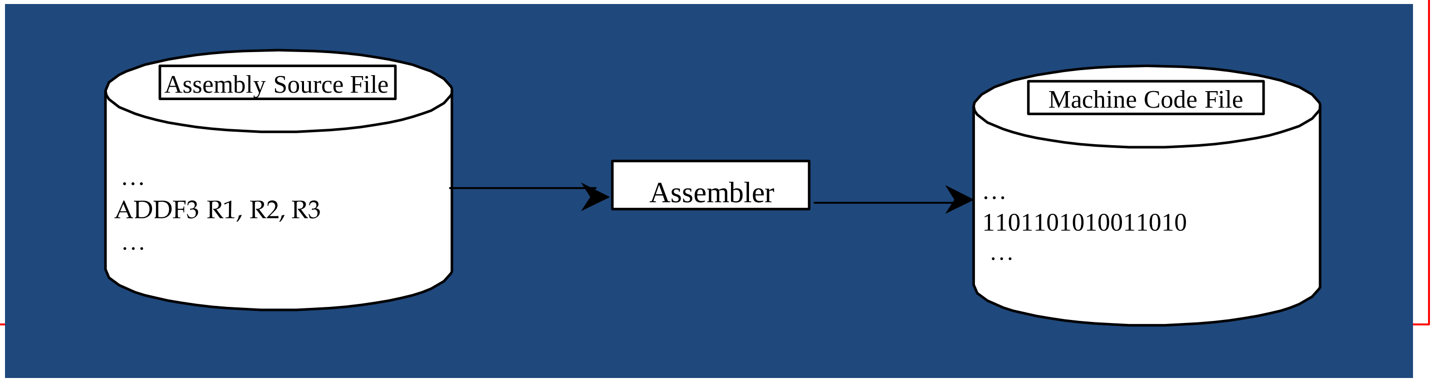
```
1101101010011010
```

Programming Languages

Machine Language Assembly Language High-Level Language

Assembly languages were developed to make programming easy. Since the computer cannot understand assembly language, however, a program called assembler is used to convert assembly language programs into machine code. For example, to add two numbers, you might write an instruction in assembly code like this:

ADDF3 R1, R2, R3



Programming Languages

Machine Language   Assembly Language   High-Level Language

The high-level languages are English-like and easy to learn and program. For example, the following is a high-level language statement that computes the area of a circle with radius 5:

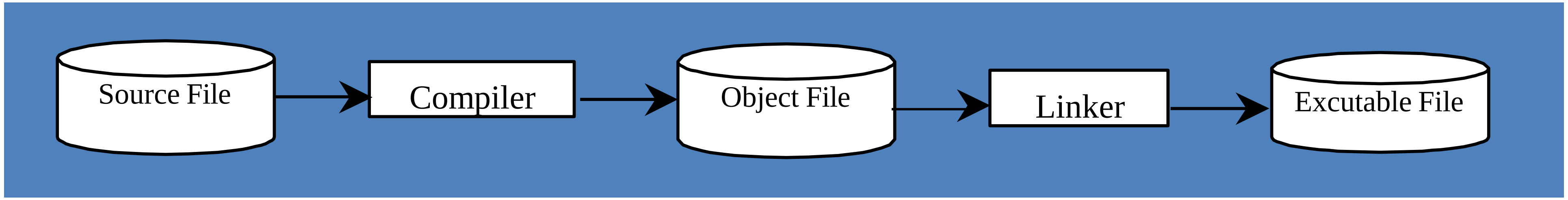
```
area = 5 * 5 * 3.1415;
```

Popular High-Level Languages

- COBOL (COmmon Business Oriented Language)
- FORTRAN (FORmula TRANslation)
- BASIC (Beginner All-purpose Symbolic Instructional Code)
- Pascal (named for Blaise Pascal)
- Ada (named for Ada Lovelace)
- C (whose developer designed B first)
- Visual Basic (Basic-like visual language developed by Microsoft)
- Delphi (Pascal-like visual language developed by Borland)
- C++ (an object-oriented language, based on C)
- Java (We will use it in the course)

Compiling Source Code

A program written in a high-level language is called a *source program*. Since a computer cannot understand a source program. Program called a *compiler* is used to translate the source program into a machine language program called an *object program*. The object program is often then linked with other supporting library code before the object can be executed on the machine.



Popular Programming paradigms

1 Structured programming techniques

2 Procedural-oriented programming

3 Functional programming

4 Logic programming

5 Event-driven programming

6 Object-oriented programming

Structured programming [Cont….]

- Structured programming techniques involve giving the code you write structures, these often involve writing code in blocks such as:
  - Sequence - code executed line by line
  - Selection - branching statements such as if..then..else, or case.
  - Repetition - iterative statements such as for, while, repeat, loop, do, until.

Procedural-oriented programming [Cont….]

- Sharing the same features as structured programming techniques, Procedural-oriented programming implement procedures/subroutines to execute common functionality

Functional programming [Cont….]

- In functional programming programs define mathematical functions. A solution to a problem consists of a series of function calls.
- There are no variables or assignment statements, but instead there are lists and functions that manipulate these lists.
- An example of a functional programming language is python.

Logic programming [Cont….]

- A logic program consists of a set of facts and rules.
- A knowledge base is built up about a specific subject and an inference engine uses the knowledge base to answer queries which are presented in the form of a goal.
- Logic programming is often used for artificial intelligence systems for example Prolog.

Event-driven programming [Cont….]

- Event Driven programming refers to your standard Windows Form idea, the program waits in a loop until an event(e.g. the click of a button, or a keystroke).
- The program then runs the code associated with this event and then returns to its loop, providing that the code did not instruct it to close.
- If more than one event occurs, code is queued by the program and run in the order the events were triggered.

Object-oriented programming [Cont….]

- Object-oriented programming takes the techniques used in structured programming further and combines routines and the data they use into classes.
- The data items stored for a class are called fields and the routines that operate on these fields are called methods.
- To use a class a programmer can declare instances of the class, which are called objects.
- An example of a Object Oriented programming language is Java.

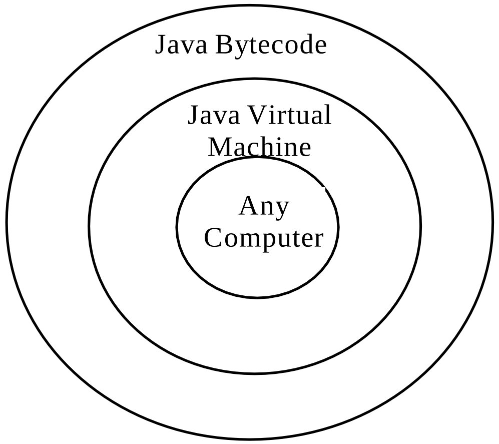
Java

- Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented,and specifically designed to have as few implementation dependencies as possible.
- It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.
- Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture.
- Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported over 9 million developers.
- Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform.
- The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

Source: https://en.wikipedia.org/wiki/Java_(programming_language)

Compiling Java Source Code

You can port a source program to any machine with appropriate compilers. The source program must be recompiled, however, because the object program can only run on a specific machine. Nowadays computers are networked to work together. Java was designed to run object programs on any platform. With Java, you write the program once, and compile the source program into a special type of object code, known as *bytecode*. The bytecode can then run on any computer with a Java Virtual Machine, as shown in Figure Java Virtual Machine is a software that interprets Java bytecode.



Java, Web, and Beyond

- Java can be used to develop Web applications.
- Java Applets
- Java Servlets and JavaServer Pages
- Java can also be used to develop applications for hand-held devices such as Palm and cell phones

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java is partially modeled on C++, but greatly simplified and improved. Some people refer to Java as "C++--" because it is like C++ but with more functionality and fewer negative aspects.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java is inherently object-oriented. Although many object-oriented languages began strictly as procedural languages, Java was designed from the start to be object-oriented. Object-oriented programming (OOP) is a popular programming approach that is replacing traditional procedural programming techniques.

One of the central issues in software development is how to reuse code. Object-oriented programming provides great flexibility, modularity, clarity, and reusability through encapsulation, inheritance, and polymorphism.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Distributed computing involves several computers working together on a network. Java is designed to make distributed computing easy. Since networking capability is inherently integrated into Java, writing network programs is like sending and receiving data to and from a file.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

You need an interpreter to run Java programs. The programs are compiled into the Java Virtual Machine code called bytecode. The bytecode is machine-independent and can run on any machine that has a Java interpreter, which is part of the Java Virtual Machine (JVM).

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java compilers can detect many problems that would first show up at execution time in other languages.

Java has eliminated certain types of error-prone programming constructs found in other languages.

Java has a runtime exception-handling feature to provide programming support for robustness.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java implements several security mechanisms to protect your system against harm caused by stray programs.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Write once, run anywhere

With a Java Virtual Machine (JVM), you can write one program that will run on any platform.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Because Java is architecture neutral, Java programs are portable. They can be run on any platform without being recompiled.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java’s performance Because Java is architecture neutral, Java programs are portable. They can be run on any platform without being recompiled.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Multithread programming is smoothly integrated in Java, whereas in other languages you have to call procedures specific to the operating system to enable multithreading.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java was designed to adapt to an evolving environment. New code can be loaded on the fly without recompilation. There is no need for developers to create, and for users to install, major new software versions. New features can be incorporated transparently as needed.