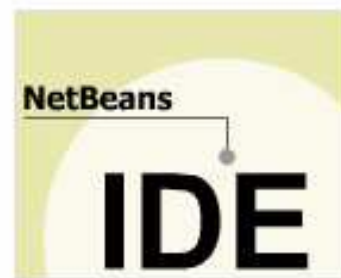


Module 1 - 2

Introduction to Java Variables and Operators

Introduction to Java

- Classes and Objects
- Getting started with Java
- Introduction to JDK
- Writing a Java program
- NetBeans IDE Overview
- Using comment in Java



Class and Object

- Describe real-world entities as objects
- Describe a software object
- Describe and explain the structure of a class
- Compare classes and objects

Class & Objects

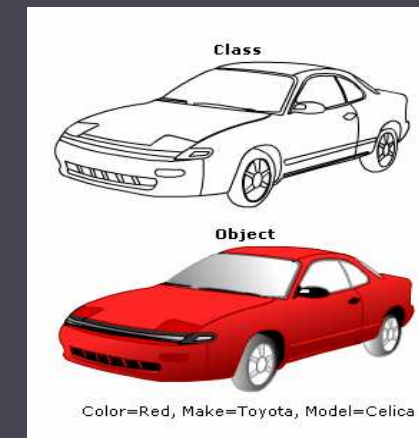
- Object is the presentation of real world entities.
- The two most important elements of object in real world are characteristics and actions

Object (Real world)	Software Object
Characteristics (properties)	State – Attributes – Variables - Fields
Actions	Behavior – Method - Functions

- Class is a template defines the outline of state and behavior for all object belonging to that class
- All instances of the class, called objects, will have common state and behavior

Class & Object: Compare

Class	Object
Class is a conceptual model	Object is a real thing
Class describes an entity	Object is the actual entity
Class consists of fields (data members) and functions	Object is an instance of a class

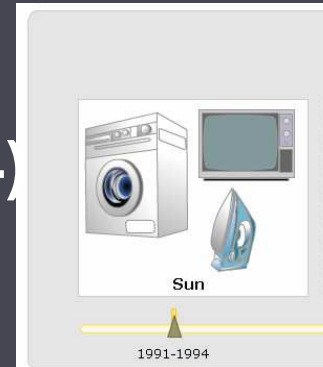


Getting started with Java

- Identify the evolution stages of Java
- State the components of the Java platform
- List the features of Java as programming language

The evolution stages of Java

- **Embedded Systems (1991 – 1994)**

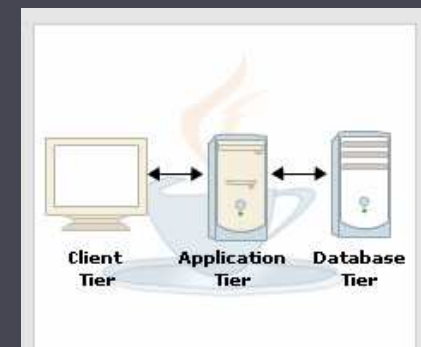


- **A client – side Wonder (1995 – 1997)**

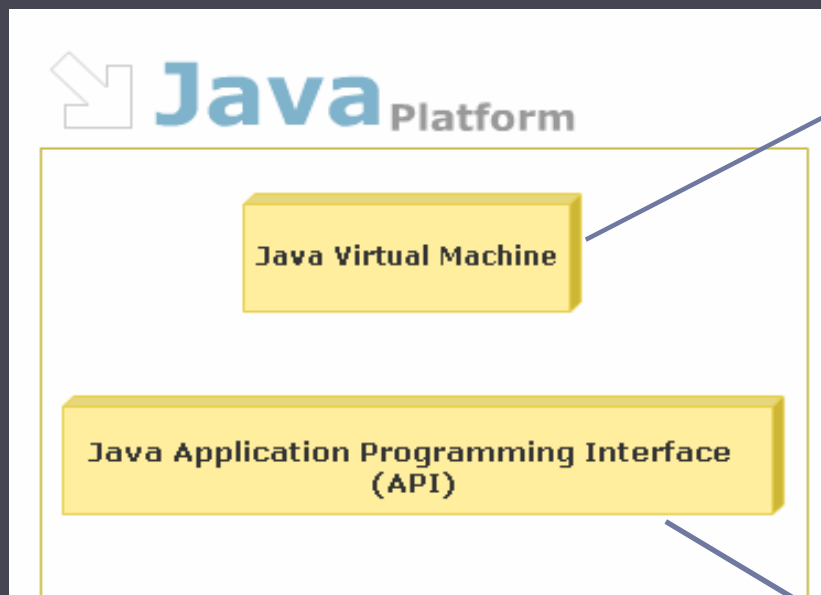


- **Moved into the Middle – tier (1997 – to present)**

- **Future: may gian more success**



Java platform



The Java Virtual Machine (JVM)

The Java Virtual Machine (JVM) is the Java runtime environment and is available on different operating systems. It serves as the intermediary between a Java program and a host computer. JVM executes compiled Java programs (byte codes). It forms a layer of abstraction for:

- Underlying hardware platform
- Operating system
- Compiled code

Different versions of JVMs are available for different operating systems.

The Java Application Programming Interface (API)

Java APIs contain vast libraries of classes and other software components such as interfaces. These are included as a part of the Java SDK. Newer releases of Java APIs provide enhanced features with introduction of new class libraries and packages.

Java – Programming language

- **Object – oriented**: The elements are present can only be accessed through classes and objects.
- **Platform – independent**: The ability of a program to run on any machine regardless of underlying platform
- **Robust**: Java requires all data to be declared explicitly
- **Secure**: Provides a secure environment and several layers of security controls
- **Distributed**: To develop application portable across multiple platforms.
- **Multi-threaded**: Java provides for multi-threading to perform many tasks simultaneously

Introduction JDK

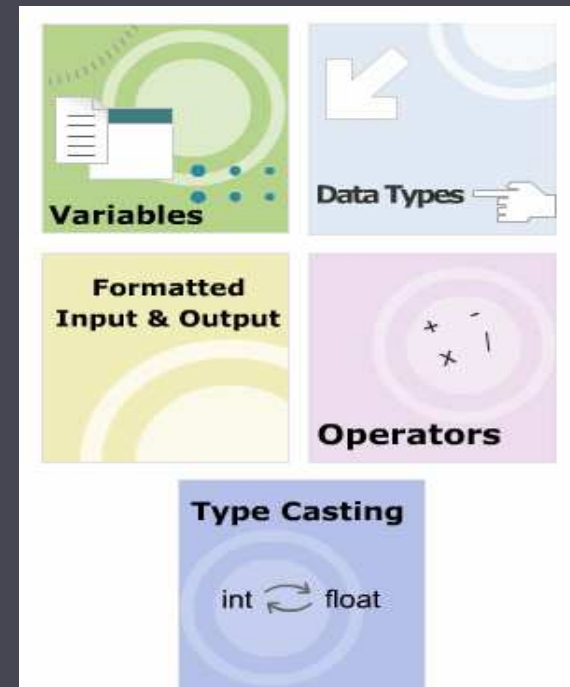
- Explain JDK and its tools
 - javac (Java compiler)
 - java (Java interpreter)
- Configure JDK
 - Path
 - Classpath
- Writing a java simple program
 - Writing
 - Compile and execute

NetBeans IDE Overview

- Identify the benefits of NetBeans Integrated Development Environment (IDE)
- Identify the elements of NetBeans IDE
- List the steps to create a Java program using NetBeans IDE

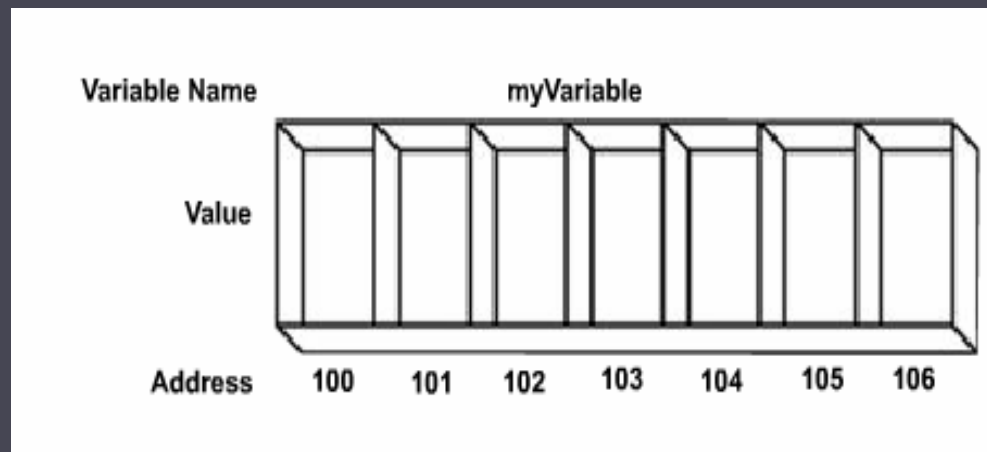
Variables and Operators

- Introduction to variables
- Introduction to Data types
- Formatted Output and Input
- Operators
- Type casting



Introduction to variable

- Variable is a location in the computer's memory where a value is stored and from which the value can be retrieved later
- Variable must be declared before they can be used



Naming convention

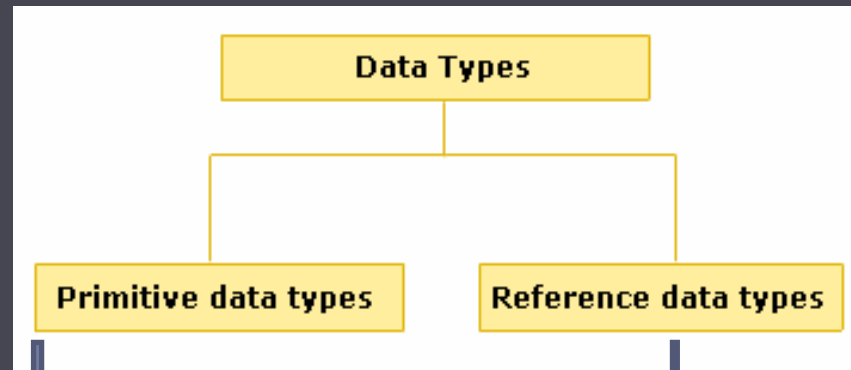
Naming Conventions



The rules and conventions for naming variables are:

- Variable names may consist of Unicode letters and digits, underscore (`_`), and dollar sign (`$`).
- A variable's name must begin with a letter, the dollar sign (`$`), or the underscore character (`_`). The convention, however, is to always begin your variable names with a letter, not `"$"` or `"_"`.
- Variable names must not be a keyword or reserved word in Java.
- Variable names in Java are case-sensitive (for example, the variable names **number** and **Number** refer to two different variables).
- If a variable name comprises a single word, the name should be in lowercase (for example, `velocity` or `ratio`).
- If the variable name consists of more than one word, the first letter of each subsequent word should be capitalized (for example, `employeeNumber` and `accountBalance`).

Introduction to Data types



Data Type	Description
byte	8-bit signed integer
short	16-bit signed integer
long	64-bit signed integer
int	32-bit signed integer
float	32-bit floating-point variable
boolean	Stores a true or false value
char	16-bit Unicode character
double	64-bit floating-point variable

Data Type	Description
Array	A collection of several items of the same data type such as names of students.
Class	A collection of variables and methods. For example, class Student containing the complete details of the students and the methods that operate on the details.
Interface	A type of class in Java used to implement multiple inheritance.

Formatted Output and Input

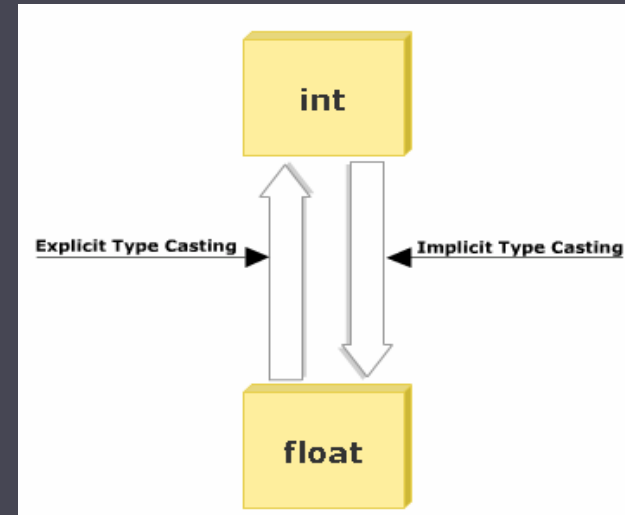
- Formtted output
- Formatted input
- List the various escape sequence

Operators

- Assignment operator
- Arithmetic – Unary operator
- Equality, Relational and Conditional operator
- Bitwise and Bit shift operator
- Operator precedence
- Operator associativity

Type casting

- State the two types of type casting



- Describe implicit type casting
 - The two types should be compatibility
 - The destination type should be larger than the source

→ the type promotion rule

- Describe explicit type casting
 - To convert a higher precision data type to a lower precision data type

Summary

Modul 1

- Classes and Objects
- Introduction to Java
- Introduction to JDK
- Writing a Java program
- NetBeans IDE
- Using comment in Java

Modul 2

- Variables
- Data types
- Formatted Output and Input
- Operators
- Type Casting