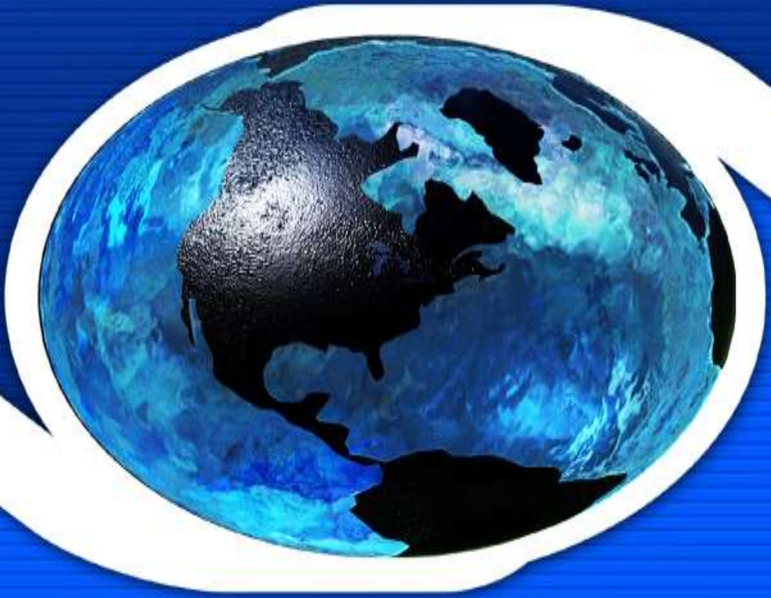
Java Programming



Introduction

- Java is a general-purpose , object oriented programming language developed by Sun Microsystems of USA in 1991.
- Originally it was called as OAK and renamed as JAVA in 1995.
- James Gosling, Patrick Naughton, Chris Warth, Ed Frank and Mike Sheridan at Sun Microsystems in 1991.
- Java was designed for the development of software for customer electronic device like TV, VCR..etc.

- Java enables users to develop and deploy applications on the Internet for servers, desktop computers, and small hand-held devices.
- Java is a general purpose programming language.
- Java is the Internet programming language.
- Java can be used to develop Web applications.
- HotJava
  - The first Java-enabled Web browser

Characteristics of Java programing

- Simple
- • Secure
- • Portable
- • Object-oriented
- • Robust
- • Multithreaded
- • Architecture-neutral
- • Interpreted
- • High performance
- • Distributed
- • Dynamic

Simple

- Java was designed to be easy for the professional programmer to learn and use effectively.
- If you already understand the basic concepts of object-oriented programming, learning Java

  will be even easier.

  Because Java inherits the C/C++ syntax and many of the object-oriented features of C++.

Secure

- Security becomes an important issues for a language that is used for programming on internet . The threat of viruses and abuse of resources are everywhere.
- Java not only verify all memory access but also ensure that no viruses are communicated with an applet.
- The absence of pointers in Java ensure that programs cannot access to memory locations without proper authorization.
- Java implements several security mechanisms to protect your system against harm caused by virus.

Portable

- Java programs are portable. They can be run on any platform without being recompiled.
- Java ensures portability in two ways.
- First java compiler generates byte code instructions that can be implemented on any machine.
- Secondly the size of the primitive data types are machine independent.

object oriented

- Java is a true object oriented language. Almost everything in java is an object.
- Object-oriented programming (OOP) is a popular programming approach that is replacing traditional procedural programming techniques.
- Object-oriented programming provides great flexibility, modularity, clarity, and reusability through encapsulation, inheritance, and polymorphism

Robust

- Java is a robust lang. it provide many safeguards to ensure reliable code.
- It provide runtime and compile time checking for the data types.
- Java has a runtime exception-handling feature to provide programming support for robustness.

Multithreaded

- Multithreaded means handling multiple tasks simultaneously . Java support multithreaded programs.
- We need not wait for the application to finish one task before beginning another.
- For ex: we can listen to an audioclip at the same time we can download applet from a distant computer.

Architecture-neutral

- Java program can be easily moved from one computer system to another anywhere at any time.
- Change or upgrade in Os , processors and system resources will not force any change in java program.
- This is the reason java used in net programming
- We can download applet from remote computer onto local system and execute it locally.

Compiled and Interpreted

- You need an interpreter to run Java programs.
- The programs are compiled into the Java Virtual Machine code called byte code.
- The byte code is machine-independent and can run on any machine that has a Java interpreter.
- Java interpreter generates machine code that can be directly executed by the machine that is running the java program.
- So java program is compiled and interpreted.

High performance

- Java performance is impressive due to the use of intermediate bytecode.
- The concept of multithread enhance the overall execution speed of java prg.
- Because Java is architecture neutral, Java programs are portable (moveable). They can be run on any platform without being recompiled.
- Dynamic
- Java is a dynamic language . Java is capable of dynamically linking in new class libraries, methods and objects.
- It is possible to dynamically link or abort the program depending on the response.

Sample java program

Enter main method

```
//This program prints Welcome to Java
 class Welcome
 {
  public static void main(String[] args) {
     System.out.println("Welcome to Java!");
   }
}
```

animation

Trace a Program Execution

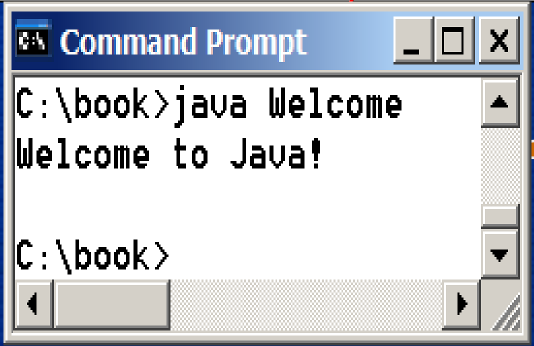
Execute statement

```
//This program prints Welcome to Java!
 class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
   }
}
```

animation

Trace a Program Execution

```
//This program prints Welcome to Java!
 class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```



```
C:\book>java Welcome
Welcome to Java!

C:\book>
```

print a message to the console

Class declaration

- The first line
- class Welcome
- declares a class, which is a object oriented contruct. Java is a true object oriented lang. so everything must be placed inside a class.
- Welcome is a java identifier that specify the name of the class.

Opening brace

- Every class defenition in java begins with a opening brace and ends with a matching closing brace.
- Class welcome
- {
- ……
- ……
- }

Main line

- The third line
- public static void main(String arg[])
- defines a method named main. Conceptually it is similar to the main() in c&c++. Every java application program must include main() method.
- This is the starting point of the interpreter to begin the execution of a program.
- A java prg. Can have any no. of classes but only one of them must include a main method to initiate the execution.
- This line include a number of keyword public, static and void

Public	**The key word public is an access specifier that declares the main method is unprotected and it accessible to all classes. This is similar to c++ public modifier.**
Static	Static declares this method as one that belongs to the entire class and not belong to the part of any object. Main must be always declares as static because interpreter uses this method before any object are created.
Void	State that the main method does not return any value.

The output line

- The only executable statement in the program is
- System.out.println(“Welcome to Java!”);
- This is similar to printf() in c or cout<< of c++.
- The println method is a member of the object out, which is a static data member of System class.
- It will print welcome to java

Three oop principles - Encapsulation

- Encapsulation is a mechanism that binds together code and data it manipulates and keep both safe from outside interference and misuse.
- One way to think about encapsulation is a protective wrapper that prevent the code and data from being accessed by other code defined outside the wrapper.
- Ex: class : A class defines the structure and behavior (data and code) that will be shared by a set of objects. Each object of a given class contain the structure and behavior defined by a set of objects.
- Switch – real life

Inheritance

- Inheritance is the process by which one object acquires the properties of another object.
- It support the concept of hierarchical classification.
- The Process of deriving new class from existing class is called Inheritance.
  New Class-Child Class
  Existing Class-Parent Class

- polymorphism:
  Poly = many
  morphs = forms
  existing in many forms.
  It is achieved by OVERLOADING and OVERRIDING.
  say a door..
  a door to a temple, a door to a house, a door to a kitty house, all are doors, but all look different.
-

Java program structure

Documentation section	$\longrightarrow$ Suggested
Package statement	$\longrightarrow$ optional
Import statement	$\longrightarrow$ optional
Interface statement	$\longrightarrow$ optional
Class defenition	$\longrightarrow$ optional
Main method class {	$\longrightarrow$ Essential
main method definition }	$\longrightarrow$ optional

Documentation section

- This section include set of comment lines, giving name of program, author, and other details.
- Comment must explain what, and why of classes and how of algorithms.

- /* ---------------------- */
- Also use // for single line

Package statement

- The first statement allowed in a java file is package statement.
- This statement declares a package name and informs the compiler that classes defined here belong to this package.
- Eg: Package student;
- The package statement is optional.

Import statement

- The next thing after the package may be number of import statements
- It is similar to # include statement is C .

- For ex: import java.io.*;
- Import student.test ;

Interface statement

- An interface is like a class but include a group of method declarations.
- It is optional.
- Used to implement multiple inheritance in java
- Class defenition
- _ a java prg may include multiple class definitions. Classes are primary and essential part of java prg.
- These classes are used to map the object of real world problems.

Main Method class :

- the main method is the starting point of java prg. The main method creates object of various classes and establishes communication between them.
- On reaching end of the main, program terminates and the control passes to the os.

Java identifiers

- *Identifiers* are the names of variables, methods, classes, packages and interfaces.
- Java identifiers follow the following rules.
- 1. They can have alphabet, digit and underscore and dollar sign characters.
- 2. They must not begin with a digit.
- 3. uppercase and lowercase letters are distinct
- 4. they can be of any length.
- Identifiers must be meaningful, short enough to be quickly and easily typed and long enough to be descriptive and easily read.

Variables

- The variable is the basic unit of storage in a java program.
- In java all variables must be declared before they can be used. The basic form of a variable is
- Type identifier[=value] [,identifier[=value];
- The identifier is the name of the varible
- To initialize the variable by specifying an equal sign and a value.
- Eg: int a,b,c;
-   int d=90, e,f=5;
-    byte z=22;

Literals

- A constant value in java is created by using a literals representations of it.
- Ex: 111 , 98.34, ‘x’, “this is a test”
- Java have 5 major type of literals
- Integer literals
- Floating point
- Character
- String
- Boolean literals

Operators

- Java support rich set of operators.
- Operator tells the compiler to perform certain
- Arithmetic Operators
- Relational Operators
- Logical Operators
- Assignment Operators
- Increment and decrement
- Conditional operators
- Bitwise Operators
- Special operators

Arithmetic operators

+	Addition
-	Abstraction
*	Multiplication
/	Division
%	Increment
+=	Addition assignment
-=	Substraction assignmen
*=,/=, %=,	------

Relational operators

- == equal to
- != not equal to
- > greater than >= greater than or equal to
- < less than
- <= less than or equal to

Logical operators

- Operation Meaning Note
- **a && b** logical AND
- **a || b** logical OR
- **a & b** boolean logical AND
- **a | b** boolean logical OR
- **a ^ b** boolean logical exclusive OR
- **!a** logical NOT
- & = And assignment
- != Or assignment
- ^= xor assignment

Assignment

- = assignment opr
- Var=expression;

- Inc/Dec
- ++ increment opr
- -- decrement opr

- Conditional opr

- ? :   if then else
- Exp1 ? Exp2 : exp3
- A=10; b= 15; x=(a>b) ? a:b;

Bitwise opr

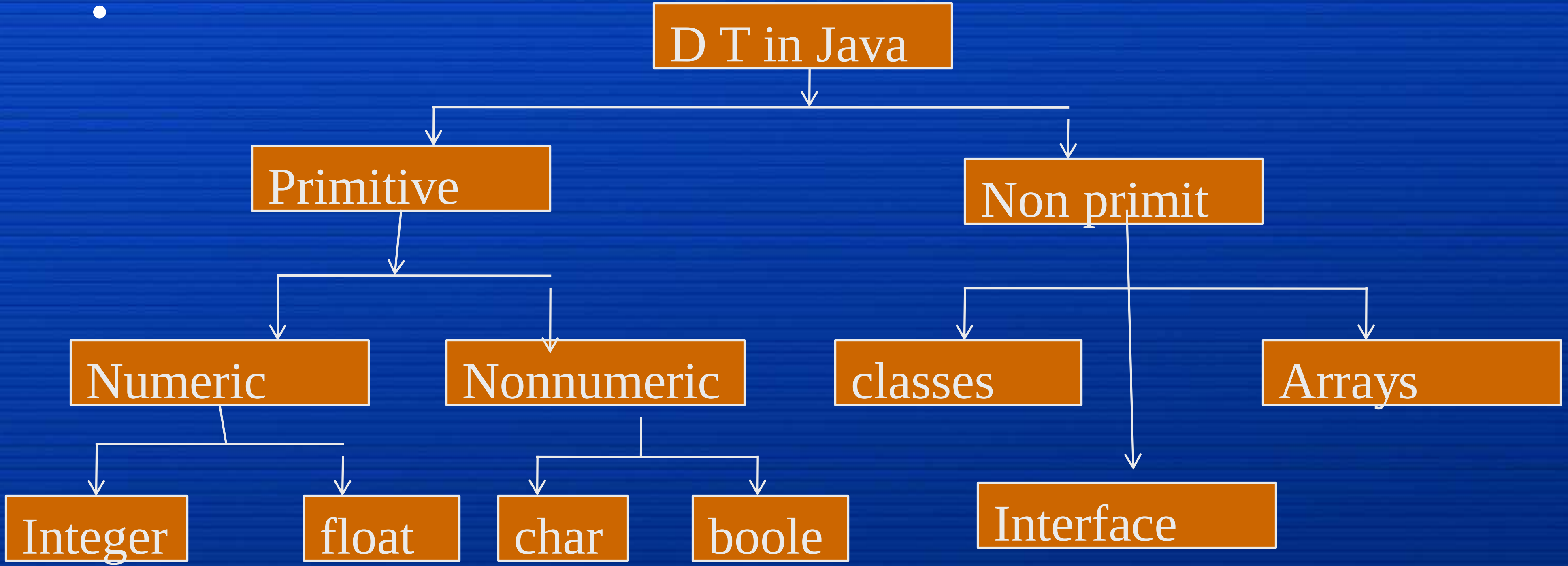
- & bitwise AND
- ! bitwise OR
- ^ bitwise exclusive OR
- >> shift right
- >>> shift right zero fill
- << shift left
- &= bitwise AND assignment
- != bitwise OR assignment

Special opr

- Instanceof operator - instanceof
- Member selection operator (.)
- The instanceof is an object reference opr and returns true if the object on the left hand side is an instance of the class given on the right side . This opr allows us to determine whether the object belongs to a particular class or not
- Person instanceof student
- Is true if the object person belongs to the class student; otherwise it is false.
- Dot opr : used to acess the instance variables and method of class object.

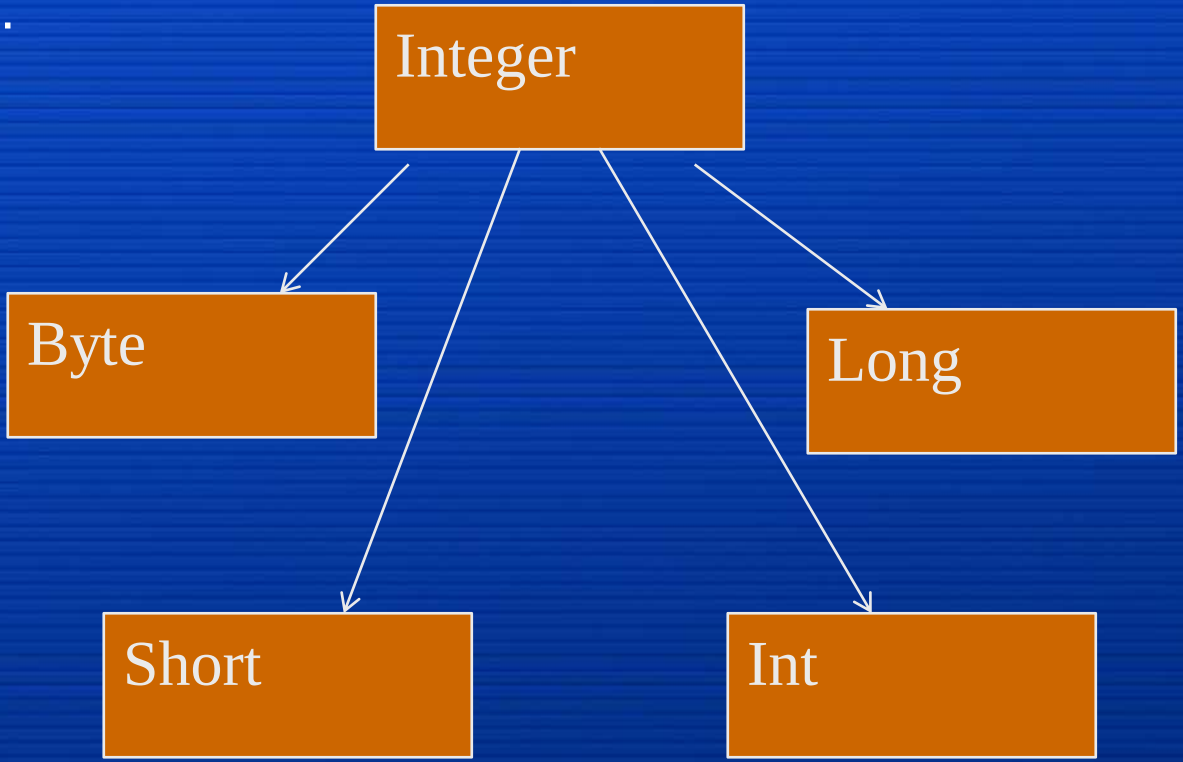
Data Types in java

- Data types specifies the size and type of values that can be stored in a variable.
-



integers

- Java defines 4 integer types : byte, short ,int and long. All these are signed,positive and negative values.



The width of an integer type should no the thought of as the amount of storage it consumes, but rather as the behavior it defines for variables and expression of that types. Java runtime environment free to use whatever size it want.

Size & range of integer

Type	Size	Max .value
Byte	One byte	-128 to 127
short	Two byte	-32768 to 32,767
int	Four byte	-2,147483 648 to 2,147,483,647
long	Eight byte	

Byte: It is the smallest integer type . The variable of type byte are especially useful when we work with a stream of data from network of file.
Also useful when working with raw binary data that may not be directly compatible with java's other built-in types.

```
byte  b,c,d;
```

- Short:- short is signed 16 bit type. It is probably the least used dt.
- short s;

- int:- most commonly used DT. It is signed 32 bit type. It is the most versatile and efficient type, and it should be used most of the time when you want to create a number for counting or indexing array.
- int mark,total;

Floating point type

- Floating point hold numbers containing fractional part .
- Eg : 45.56
- Float type value are single precision numbers
- Double type value are double precision numbers

Float	**4byte (1.8e+308)**
Double	8 byte (3.4e+038)

Character & boolean

- To store character constants in memory use char data type
- The size of char is 2 byte

- Boolean is used when we want to test a particular condition during the execution of program.
- There are only two value that a boolean type can take true or false.
- boolean a,b;

Control statements

- Java prog. Control statements can be classified into following catageries
- 1.selection statements
- 2. iteration statements
- 3. jump statements
- Selection stmt allow ur prog to choose different paths of execution based upon the outcome of an expression or the state of the variable.
- Iteration stmt enable the program execution to repeat one or more stmt.
- Jump stmt allow ur prog to execute in nonlinear way.

Selection statement

- Java support 2 selection stmt - if & switch
- If statement is java's conditional branch stmt.
- It can be used to route program execution through 2 different paths. The general form is
- if (condition)
- statements;
- else
- statements;

if else if

```
if (condition1)
      statements1;
else if(condition2)
      statements 2;
 else if(condition3)
       statement3;
  else if(condition n)
        statement n;
   else
         default statements;
```

Nested if

```
•   if(condition)
•     {
•     if(condition)
•            { statements; }
•      else
•              { statements;}
•       }
•     else
•         { statements; }
•    statements x;
```

switch

- The switch statement is java’s multiway branch statements. It provide an easy to dispatch execution to different parts of ur code based on the value of an expression.
- switch(expression)
- {
- case value1: statements; break;
- case value2: statements; break
- …..
- case valueN: statements;break;
- default: default statements;
- }
-

Iteration statements

- Java’s iteration statements are
-       while
-        for
-        do while
- while(condition)
-    {
-      //body of loop
-    }
- int n=10;
- while(n>0)
-    {
-      System.out.println(“ no “+n); n--; }
-

do- while

- If the conditional expression controlling while loop is initially false, then the body of loop will not execute.
- Sometimes it is desirable to execute the body of while loop at least once, even if the conditional expression is false to begin with.
- Test the condition at the end of loop not at the beginning.
- The do-while loop always execute its body at least once, because its conditional expression is at the bottom of the loop.

```
• do {
•       // body of loop
•       }while(condition);
```

for loop

- For loop work as follows
- When loop first starts, the initialization portion of the loop is executed.
- Next condition is evaluted.
- If this expression is true then body of the loop is executed. If it is false then loop terminate.
- Next iteration portion of the loop is executed. This is usually an expression that increments or decrements the loop control varible.

Jump statement

- Java support 3 jump statements
- break
- continue
- return
- Break statement has used to terminate a statement sequence in a switch statement. Also used to exit a loop. Break also used to provide a civilized form of goto statement.
- For(i=0;i<100;i++)
- {
- if(i==5) break;
- System.out.println("I ="+i);
- }

Using break as a form of goto

- The general form of labeled break statement is
- break label;
- label is name of a label that identifies a block of code. When this form of break execute , control is transferred out of named block of code.

- Program to display Fibonacci series.
- Program to display the series   * * * * *
-                                  * * * *
-                                  * * *
-                                  * *
-                                   *
-
- 5 4 3 2 1
- 5 4 3 2
- 5 4 3
- 5 4