Date.
Page No. 1

Introduction to Java Language

A Language in which user must know the internal structure of a computer before coding a program is known as Machine Dependent Language.

To overcome the demerits of Low level Languages, the High level languages are developed. They are machine independent and allow users to write a program by using simple instructions like English sentences. Hence, it is easier to understand and develop programming logic in High level languages.

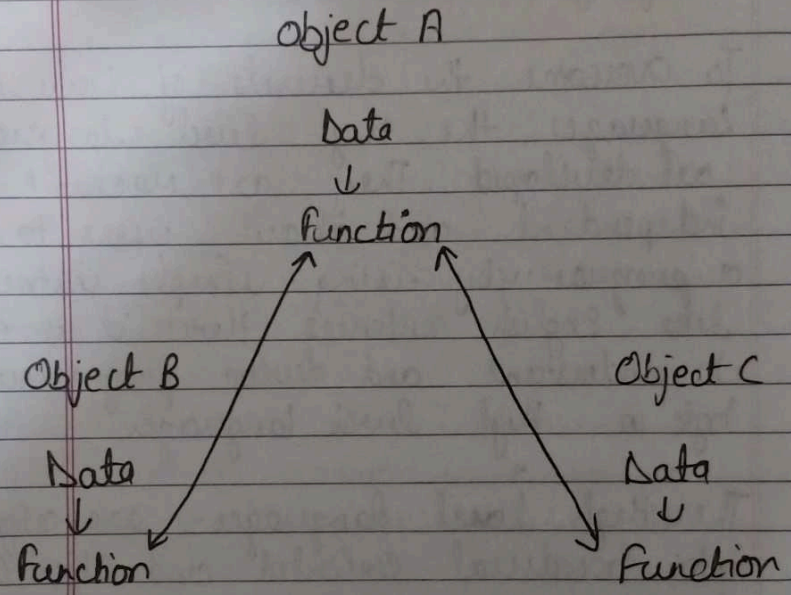
The High level languages are categorised into Procedural Oriented and Object Oriented Languages.

Procedural Oriented Language is an approach where the emphasis is on the functions rather than the data values.

Data security is a real threat in POP system.

Date.
Page No. 2

An Object Oriented Programming (OOP) is a modular approach, that allows the data to be used with stipulated program area. Under this programming approach, the emphasis is on data rather than functions.



In the above diagram, each object includes data and function separately. The data items of an object can only be operated by the function available within the same object. However the objects can communicate with each other through the functions.

Features of OOP

1. Emphasis is on data rather than functio
2. It insures data security to prevent it from getting corrupted during various opelations
3. Allows creating various objects in such a way that each object contains a set of data and functions
4. The objects can interact with each other through the functions.

Principles of OOP

1. Encapsulation
The wrapping up of a data and function of an object together into a single unit is called encapsulation

2. Data Abstraction
An act of representing essential feautres while hiding the background details is called data abstraction.

3. Inheritance.
Inheritance is the process by which objects of one class can acquire some

common properties of objects from another class.

4. Polymorphism
The process of using a function for more than one purpose in an OOP is called polymorphism.

5. Dynamic Binding.
Dynamic binding is the process to link the function call with function signature at run-time i.e during execution of a program.

Java is an Object Oriented programming language. In 1991 the Sun Micro Systems developed a complete language as a part of research work to develop software for the consumer electronics.
The name JAVA came from several individuals involved in the project
J → James gosling
A → Arthur Van Hoff
↕
A → Andy bechtolsheim

It was originally called Oak by James Gosling.

Compiler and Interpreter

The software, that is used to convert the high level instructions line by line to the machine level language is known as Interpreter.

The software that accepts the whole level instructions at one go and then converts it to machine level language is known as compiler.

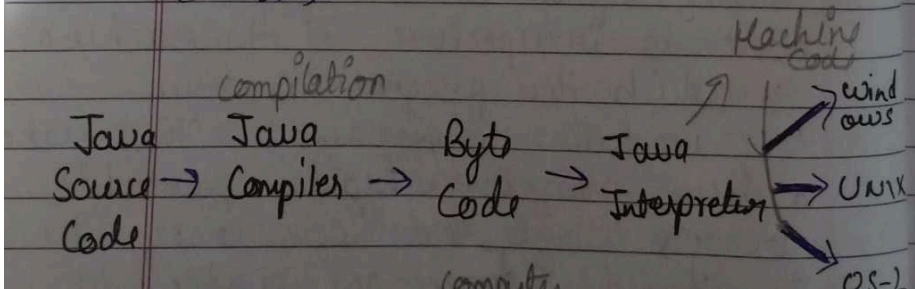
Java Compiler and Interpreter

Java compiler is used to convert a java program into an intermediate binary form called Byte code. This code is irrespective of the machine on which the program is run (i.e machine independent). This make a Java program highly portable as its byte code can easily be transferred from one system to another. When the Byte code is to be run on a specific system, an interpreter is needed that converts

the byte code to machine code suitable to the platform.
Java machine code varies for different platforms like Windows UNIX etc. Hence Java Interpreter converts byte code to the machine code for specific platform. note

Thus, the java program uses both the compiler as well as an interpreter.

Java software being a software acts as a machine that accepts byte code and generates required machine code. Hence, java interpreter is also called Java Virtual Machine (JVM)



JVM
Java Virtual Machine

JIT
Just in Time
Translate the code at the time of execution

class loader
Byte code verifier
Interpreter
JIT

Date.
Page No. 7

JDK (Java Development Kit)
It contains Java Class Library that includes various packages.

note A package is a collection of the classes which is included in a program as per the need of the user.

java.lang: to support classes containing functions to operate on String/Characters, Math, Integer, etc.
java.io : to support classes to deal with input and output operation.
java.Math: to support Mathematical function such as square roots (integer and decimal both)
java.applet: to support classes to generate applets for scientific environment
java.net: to support classes for network related operations and dealing with URL
java.awt: to support abstract window tool kit and also to manage GUI
java.txt: to use text elements such as date, times and currency etc.

URL → Uniform Resource Locator
GUI → Graphic user Interface.

(JIT)

Source Code

An application program written in high level language, which is an input to a computer system

Byte Code

The conversion of a Java program in an intermediate language is called as Byte Code

JVM

Java Virtual Machine is an interpreter which converts byte code to a machine code.

JRE

Java Runtime Environment helps in executing programs developed in java.

Java Reserved words / Keywords

Keywords are the reserved words which are preserved with system that carry special meaning for the system compiler. These words cannot be used as a variable name in any program

case	switch	else	break	static
do	const	throws	float	char
try	int	double	void	goto
for	while	new	package	private
long	if	byte	import	boolean
catch	short	public	class	default

Comments

Comments in java are the statements that are not executed by the compiler and interpreter. It can be used to provide information or explanation about the variable, method, class or any statement.

There are three ways to give a comment in Java

1. // : used for single line comment
2. /* comments */ : used for multi line comment
3. /** Comment **/ : used for documenting comment

Date.
Page No. 10

Output Statement

System.out.println() statement, the cursor skips the line and passes to the next line after displaying the required result

System.out.print() statement, the cursor remains on the same line after displaying the result

Note The message is to be written within double quotes (" ") enclosed within braces

Message along with a variable must be separated by a + sign.

Date.
Page No. 11

Objects And Classes

Java is an Object Oriented Programming language. It is called object Oriented Programming language because it follows OOP's concept.

Two more basic components of Object Oriented Programming system are Object and class.

Class

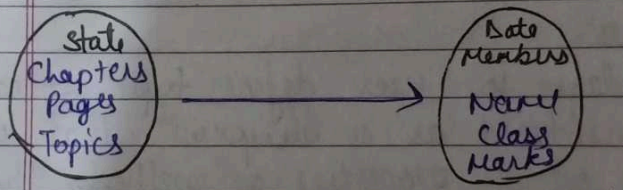
A class is user define type. It can be considered as a blueprint that represents the set of properties or methods that are common to all objects of one type

Object

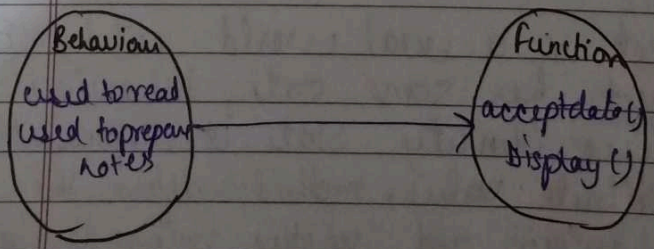
Object is a real world entity. Each object has some state, behaviours and unique identity. State is represented by attribute value, methods define the behaviour and unique name is an identity of an object.

Date.
Page No. 12

Object	State	Behaviour
Real world (book)	Chapters Pages Topics	used to read used to prepare notes
↕	↕	↕
Object Software (Student)	Data Members Name Class Marks	Functions Acceptdata() Display()



Real world object (Book) — Software Object (Student)



note
Class is an object factory
Class is user defined data types
Class is a composite data type
Object is an instance of a class.

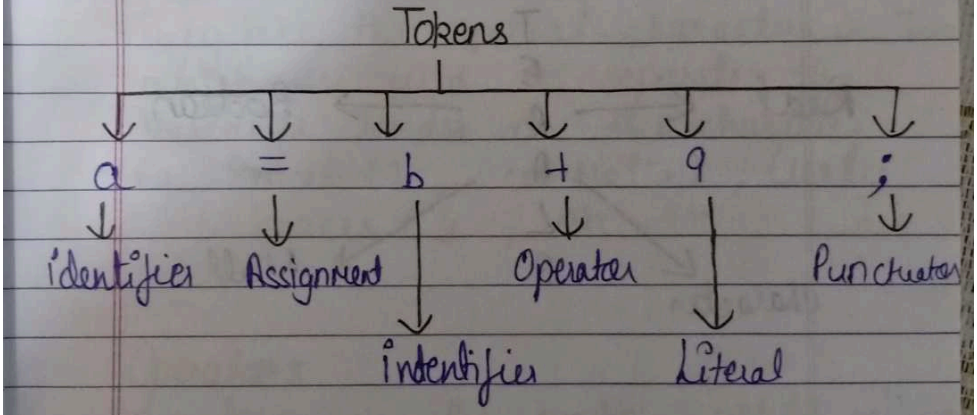
Date.
Page No. 13

Concept of Data Types

In Java, a data type allows the compiler to allot proper space in the memory for data storage.

Tokens

Each individual character used in a Java program is termed as token. It is the fundamental unit of a program



Types of Tokens

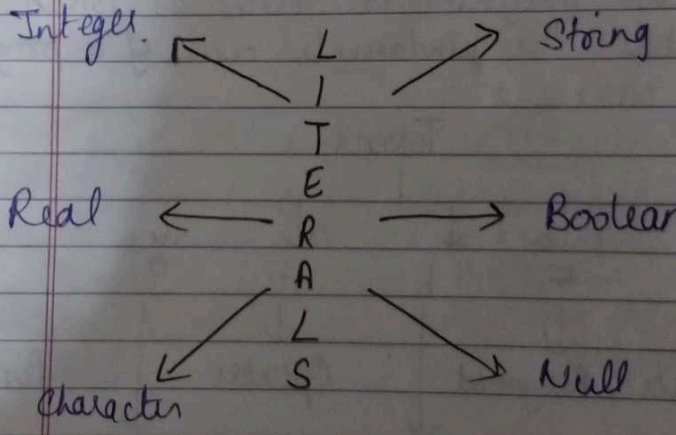
There are various types of tokens used in Java. They are

- Literals
- Punctuators
- Identifiers
- Separators
- Assignments
- Operators

Date.
Page No. 14

Literals (constants)

Literals are the constants used in a Java program. When we write a Java program some quantities remain fixed throughout the discussion of the program. Such quantities are called literals or constant



Punctuators

Punctuators are the punctuation signs used as special characters in Java. Some of the punctuators are

Dot (.) → used to represent the

scope of a function i.e function belonging to an object or a class.

Semi colon(;) → used in a java program as a statement terminator.

note Any line continued after semi colon is treated to be the next line.

Separators

They are the special characters in Java which are used to separate the variables ~~under~~ or the characters
eg Braces (), comma(,), Curly braces {}, etc.

Operators

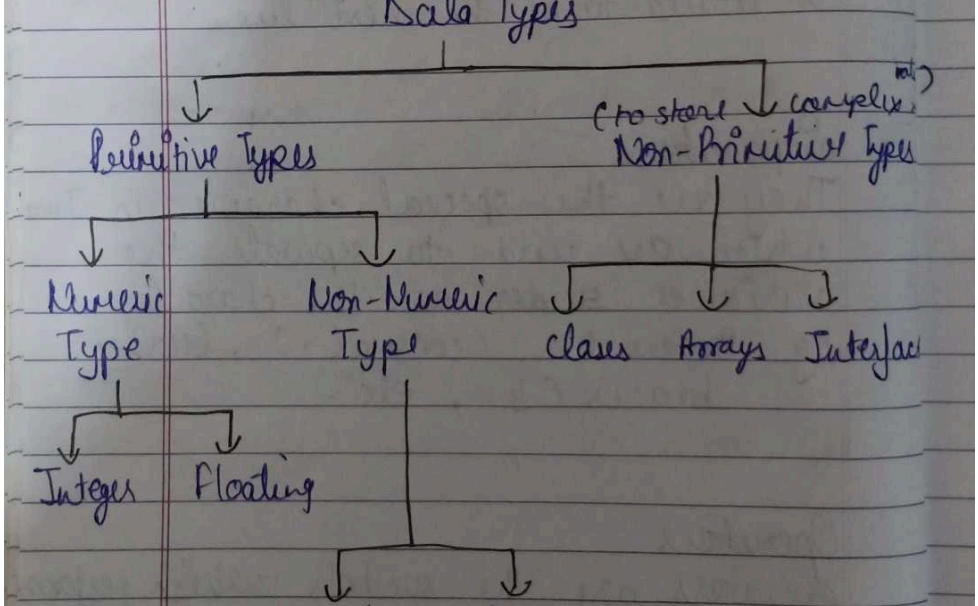
Operators are the symbols which perform arithmetical or logical operations to yield a meaningful result.
There are three types of operators

Arithmetical	+, −, /, * etc.
Relational	<, >, ==, !=, <= etc
Logical	&&, ||, ! etc.

Date.
Page No. 16

Data types

Data types are declarations for variables. This determines the type and size of data associated with variables which is essential to know since different data types occupy diff. sizes of memory.



There are two types of data types

Primitive and Non-Primitive

Primitive Types (to store simple value)
The data types which are independent of any other type are known as Primitive data types. These types are also called Basic Data Types

a) Integral Types

byte
size = 1 byte (8 bits)
Range = -128 to 127
Default value = 0

short
size = 2 bytes
Range = -32768 to 32767
Default value = 0

int
Size = 4 bytes
Range = -2147483648 to 2147483647
Default value = 0

long
size = 8 bytes
Range = -2^{63} to $2^{63}-1$
Default value = 0

b) Floating Type

float
Size = 4 bytes
Range = -3.4e38 to 3.4e38
Default value = 0.0

double
size = 8 bytes
Range = -1.7e308 to 1.7e308
Default value = 0.0.

Character Type
char
size = 2 bytes
Range = 0 to 65535
Default value = 0 (space), " "

String
size = more than 2 bytes

ASCII code
A-Z = 65-90
a-z = 97-122
0-9 = 48-57

Boolean Type
boolean
size = not defined
Range = not defined
default value = false

Date.
Page No. 19

Identifiers

Identifier is used for identification. It is used to identify user define variable, Methods, classes, interface. So identifier can be a class name, variable name, method name, interface name or a label name.

Rules to define Identifiers

- Allowed characters are
  a to z.
  A to Z
  0 to 9
  $ and _
- Start with alphabet or underscore "_" or $
  eg age12, _age, $age
- It should be meaningful
- Keywords cannot be used as identifier
- Upper and lower are distinct
  eg age, Age
- Space is not allowed

Date.
Page No. 20

- Number of symbols in identifier is not limit But lengthy identifier is not feasible
- Library methods and class name can be used as an identifier. But it reduces readability

Variable.
A variable is a named memory location that contains the value. The value of a variable can be change during the execution of program.

Syntax: <data type><space><variable name>

Date.
Page No. 21

Escape sequence

Escape sequence are the commands used along with print statement for controlling cursor on the screen in order to display information in specific format.

\t	Horizontal Tab
\v	Vertical Tab
\\	Backslash
\'	Single Quote
\"	Double Quote
\b	Backspace
\f	Formfeed
\0	Null
\r	Carriage return.

Type Conversion

A process by which, the data type gets converted after user intervention is called type conversion
There are two types of conversions

1) Implicit Type

A process by which the data type gets converted automatically into a

Higher type by defaut is called as implicit conversion.

Hierachy of Data types

byte
char
short
int
long
float
double.

2) Explicit Type
A process by which the data type gets converted after user intervention is known as explicit conversion.

Wrapper Classes
Wrapper classes provide the facility to store the primitive data values in terms of objects

long a = 5 + c.

Date.
Page No. 23

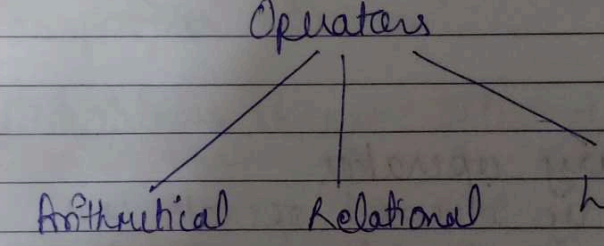
Operators

It is a symbol used to perform arithmetical or logical operations

Operands

The values which is involved in the operations are termed as operands.

Types of operators



Arithmetical Operators

The operators, which are used to perform arithmetical calculations in a program are known as arithmetical operators.
eg +, −, *, / and %.

Relational Operators

These operators are used to check the relationship among the operands

Date.
Page No. 24

eg <, >, <=, >=, ==, !=

Logical Operators

Logical operators are used to connect two or more conditions together. in a logical situation and result in collective outcome of all the conditions

eg AND (&&), OR (||), NOT (!)

Unary operator

Unary increment operator (++) increases the value of operand by one

Unary decrement operator (--) decreases the value of operand by one

Unary Increment/Decrement Operator

Prefix

Postfix

Date.
Page No. 25

Prefix
If the increment or decrement operator is used before the operand it is known as prefix operator

Principle → Change Before Action

Postfix
If the increment or decrement operator is used after the operand it is known as postfix operator

Principle → Change After Action

Q. if p = 5
find d = ++p + 5;

d = ++p + 5
= 6 + 5
= 11 Ans

Q. if a = 48 ; find a = a++ + ++a;
a = a++ + ++a
a = 48 + 50
= 98 Ans

Q If c=2; then find d= ++c + c++ + 4

d= ++c + c++ + 4
= 3 + 3 + 4
= 10 Ans

NOTE

Q If M=12; then find n= M++ *5 + --M;

n= M++ *5 + --M
= 12 *5 + 12
= 60+12
= 72 Ans

Q If y=14 then find z= (++y *(y++ +5)

z= ++y *(y++ +5)
= 15 *(15 + 5)
= 15 *20
= 300 Ans

Q If x=0 then find y= ++(++x);

y= ++(++x)

compile time error.

Conditional And Looping Statements

NOTE Some mathematical functions

1. Math.sqrt()
   Used to find square root. It returns a double type value

2. Math.min()
   Used to find minimum of two nos.

3. Math.max()
   Used to find maximum of two nos.

4. Math.pow()
   Used to find the power raised to a base. It return a double type value

5. Math.log()
   Used to find natural logarithmic value. It returns a double type value

6. Math.abs()
   Used to return an absolute value i.e only magnitude of the number. It returns int/long/double value (depends)

7. Math.rint()
Returns nearest integer value when a real number is used as its argument. It returns a double type value

eg System.out.println(Math.rint(8.5)); eg
output as 8.0

note if fractional part is more than 0.5 it gives next higher value

eg Math.rint() converts 9.5<x<10.0 eg
to 10.0

S.O.println(Math.rint(9.501));
output 10.0

8. Math.round()
Returns the value in rounded form.

eg		output
	S.O.println(Math.round(8.0));	8
	S.O.println(Math.round(8.25));	8
	S.O.println(Math.round(8.49));	8

eg
```
S.O.pln (Math.round (8.5));          output
S.O.pln (Math.round (8.75));          9
S.O.pln (Math.round (8.99));          9
                                      9
```

eg
```
S.O.pln (Math.round (-8.0));          output
S.O.pln (Math.round (-8.25));         -8
S.O.pln (Math.round (-8.5));          -8
                                      -8
```

eg
```
S.O.pln (Math.round (-8.51));         -9
S.O.pln (Math.round (-8.75));         -9
S.O.pln (Math.round (-8.999));        -9
```

Difference b/w Math.rint() & Math.round()

Math.rint()	Math.round()
It returns the closest double value	It returns the closest higher value
eg Math.rint (2.50). returns 2.0	eg Math.round (2.50) returns 3
Returns double data type	Returns long or int data type

9) Math.ceil()
Returns the highes integer of two integer no. It returns a double type value

eg
S.O.pln (Math.ceil (8)); 8.0
S.O.pln (Math.ceil (8.5)); 9.0
S.O.pln (Math.ceil (8.912)); 9.0

S.O.pln (Math.ceil (-0.45)); -0.0
S.O.pln (Math.ceil (-8)); -8.0
S.O.pln (Math.ceil (-8.912)); -8.0

10) Math.floor()
Returns the lower integer of the two integers. Double return type

eg
S.O.pln (Math.floor(8)); 8.0
S.O.pln (Math.floor(8.5)); 8.0
S.O.pln (Math.floor(8.912)); 8.0

S.O.pln (Math.floor(-0.48)); -1.0
S.O.pln (Math.floor(-8)); -8.0
S.O.pln (Math.floor(-8.5)); -9.0
S.O.pln (Math.floor(-8.912)); -9.0

Conversion from Degree to Radian

180 degree = π Radian

180 degree = 22/7 Radian

1 degree = 22/(7*180)

1 Radian = (180*7)/22

11) Math.exp()

Provides exponential value

It returns a double type value.

12) Math.random()

Used to get a numbers randomly between 0 and 1.

Use of random function

1. To get integer random number b/w 1 and n

```
int x=(int)(Math.random()*(n-1))+1)
```

2. To get integer random number b/w M and n

```
int x= (int)((Math.random()*(n-M))+M)
```

3. To obtain head or tail randomly when coin is tossed
int n = (int)(Math.random()*2)

4. To decide the points gained randomly by a player when he throws a dice having six faces
int n = (int)((Math.random()*6)+1);

note import java.io.*;

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

public static void main (String args[]) throws IOException not

* throws IOException eliminates I/O errors in the program.

System.out.println("Enter a no");
int n = Integer.parseInt(br.readline());

```
System.out.println("Enter a no");
long n = Long.parseLong(br.readline());

System.out.println("Enter a deci no");
float n = Float.parseFloat(br.readline());

System.out.println("Enter a decimal no");
double n = Double.parseDouble(br.readline());

System.out.println("Enter a character");
char ch = (char)(in.read());

System.out.println("Enter a String");
String str = br.readline();
```

note

```
import java.util.scanner;

Scanner in = new Scanner(System.in);

S.O.pln("Enter a no");
int n = in.nextInt();

S.O.pln("Enter a no");
double n = in.nextDouble();
```

```
S.O.pln("Enter a string");
String str1 = in.next();

S.O.pln("Enter a string");
String str2 = in.nextLine();
```

1. if statement
2. if and only if statements
3. if - else statement
4. if - else - if statement
5. nested if statement

Switch Case Statement / Menu Driven Program / User's Choice

The switch statement works as a jumping statement, where the control is transferred to a specific case as per the given switch value

data type used are
- numeric type
- character type
- string type

Fall through
Under certain cond. if break statement is not used at the end then the control enters into the next case statement for the execution. This unusual execution of more than one cases at a time is termed as 'Fall through'

Difference blw if-else And Switch Case

If-else	Switch Case
It results in Boolean type value	It results in int/char type value
It does not use default case	A default case is used if case is not available for a given switch value
It can perform tasks on a relational or a logical expression	It can only perform test for equality

Testing	Debugging
Testing is a process in which a program is validated	Debugging is a proce in which the erro in the program ar removed
Testing is complete when all desired verifications against specifications have been performed	Debugging is finis when here are n errors and the program is ready for execution.

Types of Errors

There are three types of errors that occur in a computer program i.e

Syntax Error
Logical Error
Run time Error

Syntax Error

These errors occur when the syntax or the grammer of the progeramming statements are not followed.

c = (a+b/2)
It has syntax error cause it doesn't have (;) so the correct form is
c = (a+b/2);

Logical Error
A logical error is an error in planning the programs logic. When logical error occurs, the computer actually does not know that an error has been made.

Run time errors
Errors which may occur other than syntax error or logical error is known as Run time error.
These errors may occur due to dividing a number by zero or to find out the square root of a negative number.

Iteration through loops

Loops

Fixed iterations:
1. for loop

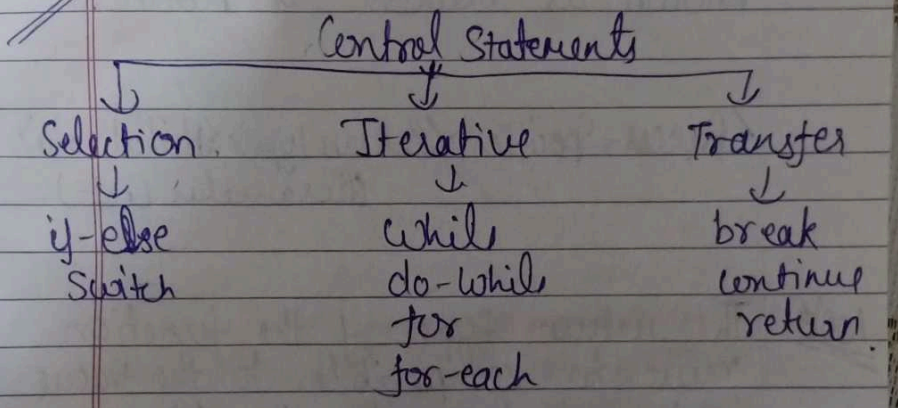
Unfixed Iteration
1. while loop
2. do-while loop

Infinite loop
Never Ending loop
eg
```
for(i=0; i>0; i++)
    S.O.println(i);
```

Difference b/w while and do while loop

While loop	Do-while loop
It is known as entry controlled loop	It is known as exit controlled loop
The loop will not execute even once if the cond. is false at the beginning	The loop will execute atleast once even if the cond. is false in the beginning

Note Break and continue statements can be used with all the formats of looping constructs for their respective purposes



Date.
Page No. 40

Methods and Constructors

A program module (a part of the program) used simultaneously at different instances in a program to perform a specific task, is known as Method or Function

<Access-Specifier><Return type><Method name>
(Parameter List)

Note: The return type of the function must be compatible to the value returned from the function.

If the function doesn't return value to the main function, then the return data type in function header should be indicated with void.

* A technique of using method in a program is referred as calling a method or invoking a method.

Actual Parameter

The parameter describes in the caller definition (the values that are passed to the method) are known as Actual Parameters

Formal Parameter

The parameter described in the function definition (that receive the values) are called formal parameters.

Pass by Value

Pass by value is the process of passing a copy of actual parameters to the formal parameters. Any change made in the formal parameters does not reflect on the actual parameters

Pass by Reference

Pass by Reference is the process of passing the reference (address) of actual parameters to the formal parameters. Any change made in the formal parameters will be reflected on the actual parameters.

Date.
Page No. 42

Pure function	Impure function
Pure function returns a value	Impure function may not return a value
It doesn't change the state of an object	It changes the state of an object
It is also known as accessor method	It is also known as Mutator method

Function Overloading

Function Overloading is the process of defining functions / methods with the same function names but with different numbers and types of parameters

note Through function overloading, Java implements Polymorphism.

Static Binding or Early Binding

The System that finds the best match of the function arguments and the parameter list during

program compilation is termed as Static Binding.

Why to use Overloaded functions?
As it is reliable to use many functions with same names for similar operations.

Recursive Function
A function designed in such a way that it calls itself in its body is called recursive function.

We can also overload main method
But JVM invokes main method.

Method Overloading
Method Overloading is a technique of defining a no of methods with same method name but with different types or number of formal parameters

Through method overloading, Java implements Polymorphism. Since polymorphism uses a method for more than one purpose

During compilation, the control seeks the best match of the types of actual parameters with the formal parameters of the overloaded methods. The method that is found suitable is linked with method call statement. This phenomenon is known as early binding / static polymorphism or compile time polymorphism

Date.
Page No. 44

Class and Constructors

SYNTAX: <Class name> <object name> = new <class name>

eg: Sum ob = new Sum();

↓ ↓ ↓ ↓

Class object new Constructor

operator

Class as an object factory

Class is a prototype of an object. Each object belonging to a specific class possesses the data and methods defined within the class. It produces similar types of objects. Hence, class is termed as an object factory.

Instance variables

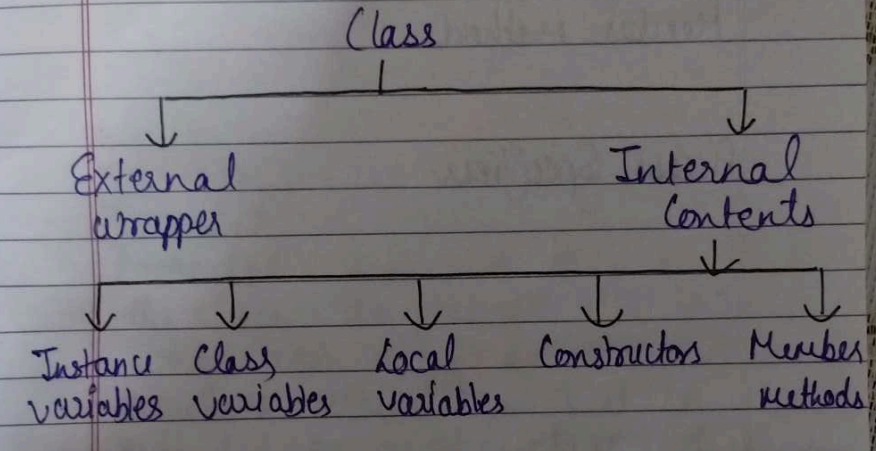
The variables, which are declared within a class are called instance variables.

Member methods

The methods which are defined within

a class are termed as Member methods. These methods deals with instance variables of the objects.

Contents of a class



External Wrapper
It is the class declaration enclosing the inner part within a pair of curly brackets

```
Class Calculate
{
  // inner part of the class
}
```

* constructor is always public. (1)

Date.
Page No. 46

It is a block of code which get executed automatically when object is created

New Operator
New operator is used to create a class object or an array.
It allocates memory space for storage of an object and returns the reference of memory at runtime.

Constructor
A constructor is a member function/method having same name as that of the class name and is used to initialize the instance variables of the objects.
(2) It does not have any return type

<Class name> <Object name> = new <Constructor>
item ob = new item()

Default Constructor
A constructor that is invoked by default and is used to initialize the data members with default initial values is known as default constructor.

Note: If we place return type before constructor name then we won't get any compilation error because it will become a normal function or method.

Date.
Page No. 49

Parameterized Constructor

It is a member function with same name as the class name which is used to initialize the object's instance variables by passing parametric values at the time of its creation

Constructor Overloading

A process of using a number of ~~function~~ constructors with the same names but different types of parameter is called as constructor Overloading.

This keyword in Java

The object on which the method is called can be referred in the method with this keyword. In other word, the keyword 'this' works as a reference variable for the current object

Application

1. It eliminates ambiguity b/w instance and local variables

byte → short → int → long → float → double
char → int

Date.
Page No. 50

1. The keyword which makes class members accessible outside the class is called public

2. The parameterized constructor is defined along with parameters

3. A class without a constructor uses default constructor

4. The constructor is used to initialize the data members of a class

5. An access specifier that prohibits a class member being used outside a class, is referred to as private

6. A constructor always has the same name as that of its name class

7. A class object is known as an instance variable

8. A copy constructor copies initial values from one object to another.

Date.
Page No. 51

9. private data members can't be accessed through objects of a class

10. A number of constructors having the same name and with different types of parameters is known as constructor overloading

11. The compiler defined constructor is known as default constructor

12. If a class doesn't indicate its modifier then it is considered to be default

Constructor	Member Method
It has same name as class name	It has different name as that of class name
It need no return data type	It uses a return data type
It always declared to be public	It may be public, private or protected
A class using a no of constructors are automatically overloaded	It needs to be overloaded

Inheritance

Inheritance is a technique by virtue of which a class acquires the properties and features (i-e Data members and member methods) from another class

A class that gets inherited to another class is known as Base Class, Parent Class or Super Class where as the class that inherits a base is said to be Derived Class, Child Class, Sub Class or Target.

A keyword 'extends' is used to inherit a base into derived class

note The access specifier of derived class is always public

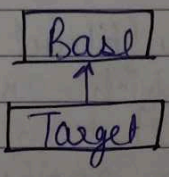
A derived class declared without any specifier is public by default.

Java uses only public mode of inheritance.

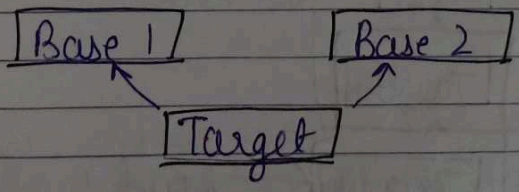
note a class member without any specifier is by default public

Types of Inheritance

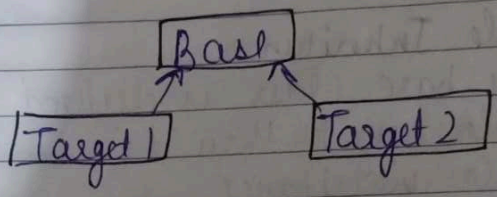
1) Single Inheritance
If a base class is derived by a single target then it is known as single inheritance



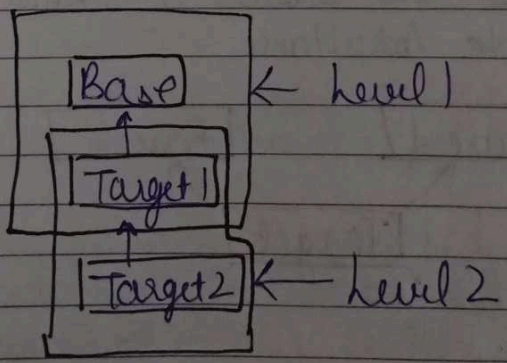
2) Multiple Inheritance
When a sub class inherits from multiple base classes, it is known as multiple inheritance



3) Hierarchical Inheritance
When many targets inherit from a single base class, it is known as hierarchical inheritance



4) Nested or Multilevel Inheritance
A target which inherits a base class can in turn be used as a base for another target. Such system is known as multilevel inheritance



Date.
Page No. 35

Method Overriding

A technique by virtue of which the methods are defined with the same name in base class as well as derived class is known as Method overriding

Static Keyword

Static keyword is used to associate a method of a given class with the class rather than the object. Static method in a class is shared by all the objects.

Static Data Member.

Static data member is also called as class variable.
class variable are also initialized with default initial values similar to instance variables

Static Member Methods

When Method is declared with static keyword, it is known as static method. Most common example of a static method is main() method. Any static member can

be accessed before any objects of its class are created, and without reference to any object.

Methods declared as static have several restrictions

- They can only call other static methods
- They can only directly access static data
- They cannot refer to this or super in anyway.

Note A static method does not modify a non-static data member where as, a non-static member method can modify the content of a static data member

The non-static data can never be accessed in static member method.

Method Overloading	Method Overriding
A no of Methods are defined with same name in a single class	A no of Methods are defined with same name in base as well as sub class.
The overloaded possess different types of parameters	The overridden method may possess similar type of parameters
It is an eg of compile time polymorphism	It is an eg of run time polymorphism
In this return type can be same or different	In this return type must be same

Abstract Keyword

Abstract Method

1. Abstract method is method that does not have an implementation or body

2. To declare abstract method we have to place abstract keyword befor method signature

3. Instead of curly braces, an abstract method will have a Semicolon(;) at the end

4. Eg abstrat void show();

OR

Abstract method is a member method that is defined with the keyword abstract and does not contain any statement within its block to execute. Such method is overridden while extending the super class into the sub class.

note

Abstract Class

Abstract class is a class that is defined with the keyword abstract and does not allow the creation of its object

Date.
Page No. 59

It is not compulsory to declare abstract method inside abstract class i.e if a class does not contain any abstract method then still we can declare the class as abstract class.

If a class contains at least one abstract method then compulsory we should declare class as abstract. Otherwise we will get compile time error

Abstract class can contain abstract method as well as non abstract method.

Even we can take constructor inside abstract class.

Note usage of abstract and final keyword at the same time is not valid

final modifier is applicable with class variable and method.