

AJT = Advanced Java Technologies

Page _____
Date _____

=> Object :

- Object are Context based and Refferable elements in real world.

=> Class :

- Class is structure and contains methods & attributes.

ASCII = 7 bytes (C, C++)

UNICODE = 2 bytes (Java)

// SAV = Smallest Allocation Unit

// MDR = Memory Data Register

// MAR = Memory Address Register

- Java uses big endian format type.
- Least Significant bytes goes to storage lowest address we refer to small endian (e.g: Motorola)
- Most Significant bytes goes to highest address we refer to big endian. (e.g: Java)
- Whenever program is want to get info outside of program we use do... while & while. Switch take always int value.
- Breaks terminate the block whereas exit(0) terminates. (Normal termination)

II NLS - National Language Support

=> Hungarian Notation:

First character of Word is Capital.

Example: Init (up : Hello World

=> 5 important OOP Concept:-

- 1.] Encapsulation (Data Hiding)
- 2.] Inheritance (Reusability)
- 3.] Polymorphism (Method Overloading & late binding)
- 4.] Abstraction (Generalization)
- 5.] Aggregation (Containment)

04/07/24 - Attributes contains raw values whereas methods contains processed values.

Character = ASCII Unicode = UNICODE

- Difference between Translation & Transliteration
 (Transliteration)

Primitive Datatypes

- int
- char
- byte
- boolean
- float
- long
- short
- double

Wrapper Classes

- Integer
- Character
- Byte
- Boolean
- Float
- Long
- Short
- Double

Note: Wrapper classes contains methods.

=> Package : Collection of related classes.

- Difference between - available & accessible

Available : Member of Parent class are available in Child class.

Accessible : Member of Parent class like private member are not accessible by Child class.

→ IS-A relationship : (Overriding)

Whenever one class inherits another class.

→ HAS-A relationship : (Not Overriding)

Whenever an instance of one class is used in another class like addressing.

- Class is user-defined or abstract datatype (Not primitive datatype). Data is user-defined datatype.

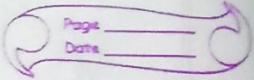
=> Aggregation :

Whenever class is contain some other class information than we call it aggregation. All the aggregation are available in a sub class.

=> Access Specifier :

- 1. Public 3. Protected
- 2. Private 4. No Access Specifier

Shuum Series = Crottfrick



05/07/24

- Object are created statically. References are by Object.
References ← Need ← Object.
- Without reference Object are not referable.
- Static member are.. allocate memory only once. Static can be variable, methods, block.
- Non - Referable Objects are removed by Garbage collection.

Note: Pointers are harmful that's why they're not in Java.

OS Area: System Space or Kernel Area
User Area = User Space

- Using pointers we can create dynamic objects.
- In-line Storage = Static creation of object
- External Storage = For making pointer we use it. Dynamic memory allocation in External Storage.
- Pointer datatype is int type (2 byte). Datatypes association with pointer Variable is a datatype of variable of reference variable.

Note: Sometimes return types of methods doesn't matter.

Formal Parameter

```
int sum(int p,int q)
{
    return p+q;
}
```

Actual Parameter

```
C = sum (20,50);
```

09/07/24

=> Method Overloading & Late Binding
(Early Binding) (Method Overriding)

- Late Binding : Based on object parameter, methods are call.

c++ Without virtual keyword method of base class is call.

- Object are bind at runtime is called Dynamic, Late, Runtime.

10/07/24 "Reference of base class will be assign object of the sub class."

In attribute case : Object of class that attribute value is refer.

- In case of dynamic calls object decide the method being call whereas the reference decide attribute being call.

=> Dynamic call or Late Binding or Runtime Binding
Jump to instruction where pointer is pointing.

=> Object Slicing:

Additional members define in child class are not callable using the reference of parent class. Only overridden method is called.

=> Cohesions:

Small precision is considered.

=> Abstraction (Generalization):

- Something that's not exist in real life.
- Pure Virtual Function
- Object can not be instantiated.
- 'abstract' keyword is use for abstract class and abstract methods.
- Non abstract method is overridden in sub classes.

11/07/24 // Learn public static void main (String args[])

Example: class HW {

 public static void main (String args[])

}

 System.out.println("Hello World");

}

3

Default Access Specifier: In Java: No access specifier
In C++: private

- ➔ main() method inside class can not call by OS because of private threads why this method defined as public.
- Note: For memory management we don't need to import unnecessary header file.

- Compiling package: `javac -d . PI.java`
 This command making directory of PI (Package) and compile all class inside it and 'Create.java' class file. Package is collection of related classes.

=> Public :

- The loading of class only creates whenever object is created.
- Within same package protected member behave like public. Outside package protected member inherit only two class.
- No access specifier can accessible within package but outside package they are not accessible.
- Object class is base of all classes.
- Default package: java.lang.*
 [Don't need to explicitly import]

=> Static :

- Static means single copy only.
- Non static member will be allocate space when object is created (They'll come into existence) whereas static member will be allocate memory irrespective of whether object of class is created or not.

- static member created before the execution of main begin.
- Static member will be invoke even without object refer by class name.
- main() is call by OS without object name that's why it's static.
- Changes done by any object of class can reflect to every object of class.

Q Why Java is robust?

Ans Java has Strong exception handling.

=> void :

void() return type just because of not return any numeric values.

=> main():

Name of the method.

=> String args[]:

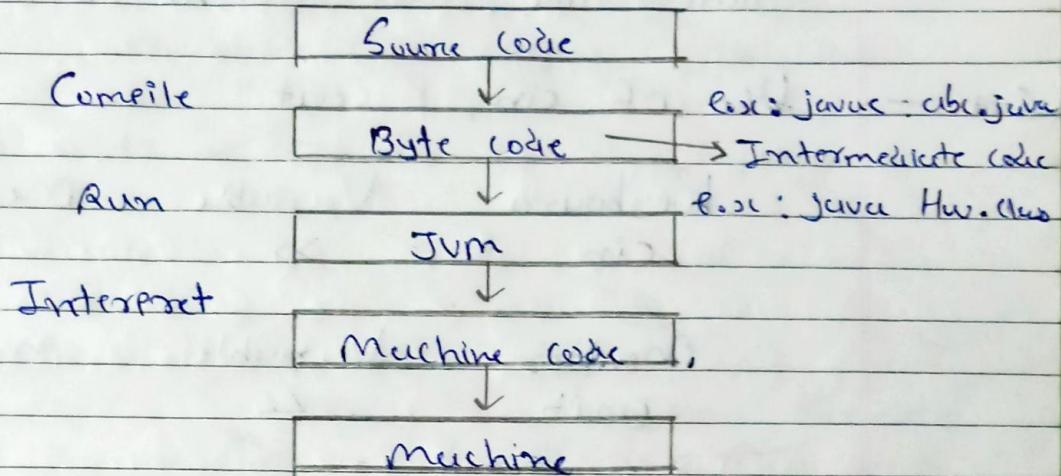
For command-line arguments, array of characters.

13/07/24

- static block is outside the methods.
- Static block will be executed first even with main() method.
- Command-line argument aren't pass.
- String class is final type.
- Java 21 Remove the public static void main() method.

Q Why platform independent?

Ans:



- Java is two phase translations. First Compile then Interpret.
 - Byte code + JVM = Java make platform independent.
 - Note: JVM is platform dependent.
 - Source code, Compiler, Bytecode are platform independent.
- => JDK = Java Development Kit (For Development)
= It contains JRE and JVM.
- => JRE = Java Runtime Environment (For run^{only})
= For get information.

C++

- cin & cout is object which are in iostream
 ↓ ↓
 ifstream ofstream header file.

// Binary computation

// Driver class
independent

&

Quantum Computation

=> Operator overloading

Date, Time, Year, Height, DOB

(++ => Use of cin & cout?)

Keyboard Variable Memory
cin >> a ;

Monitor Variable Memory
cout << a ;

=> POJO = Plain Old Java Objects.

=> System.out.println();
↓ ↓ ↓
Class Object method

System.out

↓

Class object

// Out cmd in one object.

- Infstream & printstream are static
members of class System.

=> Package Share Via JAR
 // Java Archive
 jar -cvf myapp.jar *.class

c = Create

v = Verbose

f = Files

B - jar Come with JDK.

- jar file are in compress format.
- It can be one any file Not only class file.

→ Uncompress :

jar -xvf myapp.jar

x = Extract.

→ Default Path ;

Set classpath = %classpath% ;

[Already Set classpath are given]

D:\Set classpath=j:\classpath\.;(1:javaproj\myapp.jar;



→ Setting Path in way :-

1. Set classpath = %classpath%;

2. Set classpath = ;d:\javaprogram\;

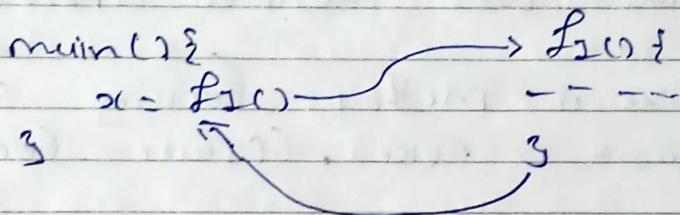
3. java -d1 c:\javaprogram\pl.A;

// classpath Setting Should refer to the Parent directory. You can not do it under getting inside directory.

★ Multithreading

- Whenever there are some processes which are sharing ^(Resource) Space So its require to create New OS multithreading.
- Two or more process sharing Resources and Parallel execution is happen.
- Multithreading is feature of OS not a programming language.
- Why programming language get multithread feature? Because Some time Read or Write is not perform only one operation at a time. (Application should have this mechanism).
- Locking mechanism also use in application development.
- Whenever You are making any API You can make class behave like threads.
- Two way to implement threading
 1. Extends Thread (class)
 2. Implement Runnable (interface)

- Java doesn't support multiple inheritance that's why we always use malleable interface.
- java.lang.* contain malleable ans threads.
- Point of call & Point to instruction.



- In join() method we use whenever function return value is not come until. (process busy execution)
- Main thread & child thread
- main() is parent of or master of all threads. Till all child threads do not complete its execution main do not ends.
- Whenever multithread environment can share resources, core allocate to the parent and child threads.
- Till the completion of child threads process main threads do not ends.

- Daemon threads; Background Processes
bc: daemon is daemon thread.
- Without explicit call daemon threads
are not come to limit.
- Example: Railways lone.
- Java uses monitor approach.
- make a method locking synchronized.
Keyword is used. (Create lock only)
- wait() & notify() methods (call
inside synchronized methods only) just
it run implicitly.
- sleep() method time is known.
milliseconds.
- When the time is not known we
Should use wait() and notify()
method.
- Joins is between Parent and child
threads.
H. join(); // wait for +1 thread

=> Runnable Interface :-

- Shared resources can modify their methods become Synchronized.
- Package Program (Follow the Rules)
- When You're doing multi-threading Programming You Need very Specific use case.
- This is always referring to the current thread. (main)
- ~~Run~~ Run from thread class Never invoke implicitly. It's Need to explicitly call.
- Normal Priority = 5 } Thread
Highest " = 10 } Priority
Lowest " = 1 }
- Meth. random() return Double type value.
- Bruce & Leander Example.
 1. Competitor Class:
Many threads implement Runnable.
 2. Resource Class:
Synchronized methods, call wait() & notify().
 3. Drive class:
main method contain only.

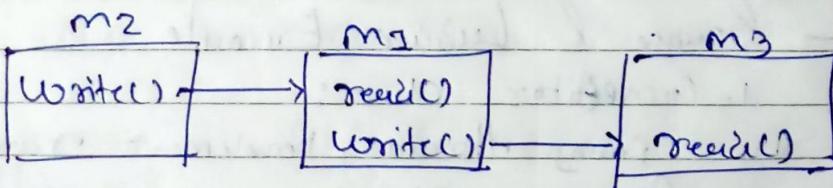
\Rightarrow I/O :-

Packages : java.io.*;

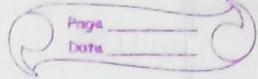
Others : java.nio.*; } For
java.nio.*; } native purpose

- 4 Basic Classes in Java :— (Belongs to java.io)
- (reading) 1. InputStream class { Byte buffer Stream }
 - (writing) 2. OutputStream class }
 - (reading) 3. Reader class { Character buffer Stream }
 - (writing) 4. Writer class }

- Character buffer Stream perform only one operation. (Element by Element)
- Byte buffer Stream perform byte by byte operation.
- InputStream class is for Read audio, Video.
- Chaining of Stream is Possible in Java.



// m1, m2, m3 are Java Programs.



Inert Stream	Output Stream	Reader	Writer
Reach	Write	Reach	Write
I/P	O/I	I/P	O/I

- Combination of Source and Destination is possible in Java
Example: Taking input from Keyboard and Writing to files.
 - We always referring to Stream, we're not directly referring to Source.
 - All the Stream can be change to each other, It can be connected to another Stream.
 - End of Stream : Stream Exchanted.
 - All 4 basic classes are abstract classes.
 - Methods in InputStream and Reader class:- read(), skip(), available(), close()
 - Signature of read() method:-
public int read() throws IOException
" (byte b[]) "
 - Partial Stream ← " ("", int b, int offset)"

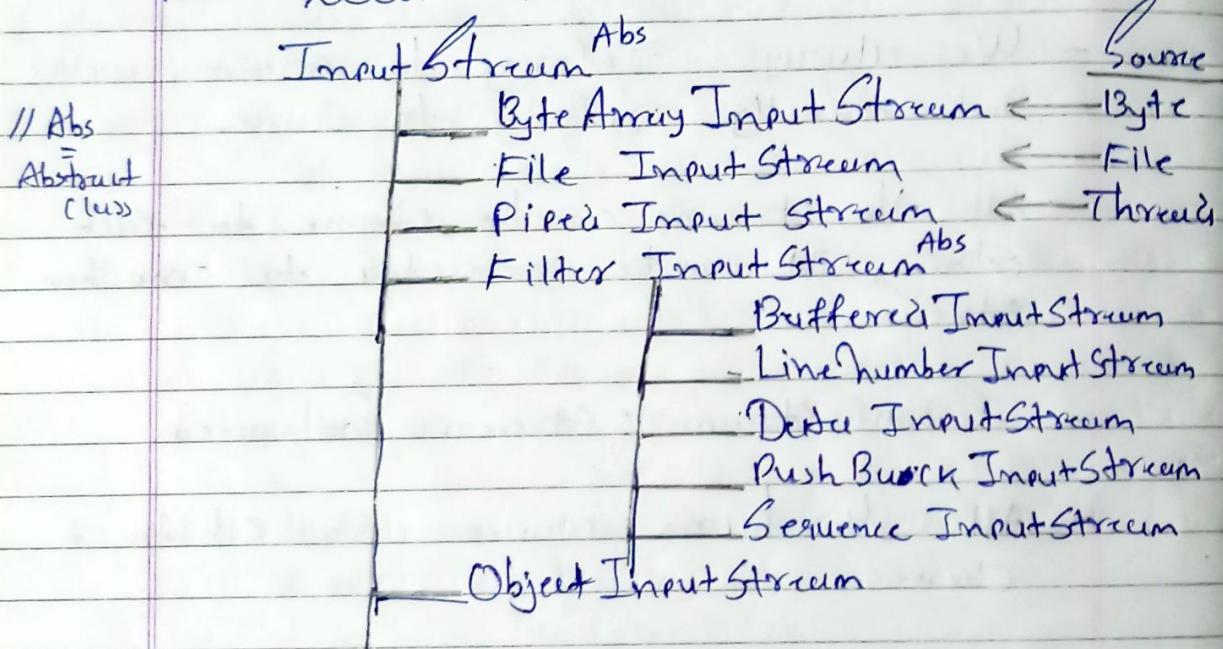
// throw = Default class exception.

// throws = Explicitly handle exception.

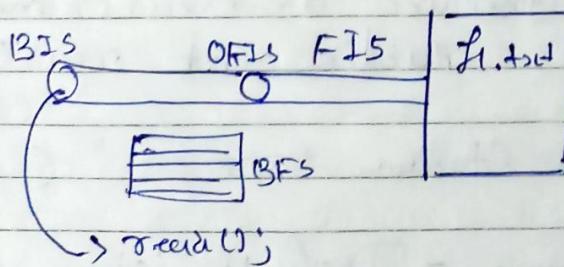
// Checked Exception is also known as
Compile time exception.

- `read()` method is also known as blocking method.
- -1 is a terminating condition (refer to end of the stream. (`read()` return value))

→ Derived classes :-

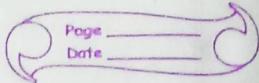


- Whenever You want to use Primitive datatype we use `DataInputStream`.



- readUTF() & readInt() methods.
- LineNumber ⇒ reader class.
- InputStream class for reading, OutputStream class for writing when you have mix data.
- unread() method for track writer position.
[PushbackInputStream]
- mark() method.
- PushbackInputStream:-
unread() method
- SequenceInputStream:-
Whenever your application needs multiple ~~Type of Source~~ sources you should use SequenceInputStream.
- ObjectInputStream:-
Parent class of all the Parent Interface Object Input.
(Contain in java.io.)
- Java Documentation. (Study Package)
- readObject() method allows you read entire stream in single operation.

Object class :- clone() method



→ Only Serializable class can read object via inputstream class.

Note: → Using object inputstream class, You can read only those object whose class is Serializable.

→ Serializable Class:-

class Student implements Serializable
{

// marking Serializable class.

3

→ Serializable is a interface which are in 'java.lang.*'.

→ It's also known as父接口父接口.

→ Cloneable is also member interface.
clone the Object.

Serializable → Convert the object into byte array
Interface and doing transmission.

Object [
Convert]

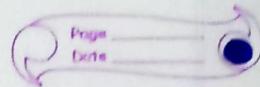
Sent → Serialize } two activity
Receive → Deserialize } mainly work.

→ There's a class Name class name by
Java, [which contain information
what you want].

01/08/2024

- 'class' is a metadata contain information.
- There's terminator in OutputStream.
- File class is not for reading and writing.
It's just for physical file.
- FileInputStream(File object)
- File class is reference to file which contain methods to verify all the files.
- isexists(), canwrite(), canread() { Validation Perform on File. }
- reader & write operation can be only Stream only.
- New line store two byte.
- ByteArrayOutputStream (ByteArray Object)'
- getBytes() method are in String class to convert String into Byte Array.
- ^{new} BufferedReader (New InputStream Reader (System.in)) [Direct mechanism]
- InputStreamReader & OutputStreamWriter are only class which are byte based Stream and character based Stream class.

Analog Signal = Byte Flow



=> SequenceInputStream:

Use for more than one sources.

=> LineNumberInputStream:

LineNumberReader() class is use.

Note: Whenever You Create Stream object
Explicitly Its Your duty to Close all
the Stream.

=> DataInputStream:

To read primitive data values. Optimize transmission time.

- Object is reads entire Objects at a time,
and can Read objects only written by
Object OS.

03/08/2024

rwS = read; Write; Synchronize

// Synchronize = Auto-Save; Simultaneous Behaviour.

=> PipedInputStream:-

Communicating between threads,
Generating Stream between two threads

// read() is a blocking method.

// Note: Object are Pass by call by reference
In Java.

05/08/2024

⇒ JDBC :- (Java Database Connectivity)

- JDBC is a driver API.
- java.sql.* ; (Package)
- - 1. JDBC ODBC Bridge (Type 1)
 - 2. Native API Driver (Type 2)
 - 3. Network Protocol (Type 3)
 - 4. Thin Driver (Type 4)

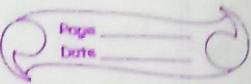
} Four main Type of Drivers.
- - 1. Unstructured Storage
 - 2. Semi-Structure Storage
 - 3. Structure Storage

} Types of Data Storage.
- - i = internet enable
 - g = grid enable
 - c = cloud enable

} (Database Software) Version.
- //ODBC = Open Database Connectivity.
- client dependent code in Native API Driver.
- Pure Java code in Thin Driver (Type 4)
- - JDK Version
 - Database
 - Database Version

} Driver Change According to it
- throws SQLException [For handling exception]

// National Language Support = NVARCHAR(10)



- Create, CreateUpdate, ExecuteQuery.
- Public, boolean, => Signature of Query
- getInt() & getString() methods.
- Statement object is local object.
- Dynamic SQL Statement.
 - ⇒ Properties() class :-
Multiple value can pass using Properties class.
String PropertyNames() method when you don't know the property key.

→ Prepared Statement => Dynamic Statement
Callable Statement => Store Procedure

Method => PreparedStatement
Class => PreparedStatement

- Store Procedure is a Database Object.
- " " " is a callable code.
- " " " has a store object that are Pre-Compiled.

Syntax :-

CallableStatement stmt = con.prepareStatement("call pl(?)")

C:\{call pl(?)}

//rw() & longrw() datatypes

Oracle Database 8
Page _____
Date _____

- registerOutParameter() = For mapping to backend to Java code.
- java.sql.Types.*; (contain final types) where Types is a interface.
- BLOB() & CLOB()
- setBinaryStream() & setAsciiStream() methods are important methods.