**Day 1 : 10-03-2021**

**Exception Handling :**

Exception is a object(memory) which occurs when unexpected or abnormal condition occurs during the execution of a program.

Java Program

javac Demo.java java Demo

java compiler java interpreter

compile time error run time error

syntax error

or

typo error

**run time error**

Error Exception

Both are pre-defined classes part of lang package.

**Error :** It is a type of run time error which we can’t handle it.

JVM Crash, Out of memory, software or hardware issue.

**Exception :** It is a type of run time error which we can handle it.

Divided by zero.

Index out of bound exception

Null pointer exception

Object

Throwable extends Object

Exception extends Throwable

Checked exception unchecked exception

RuntimeException

extends Exception

IOException ArithmeticException

FileNotFoundException ArrayIndexOutOfBoundsException

SQLException NumberFormatException

NullPointerException

Etc etc

All unchecked exception extends RuntimeException class.

But all checked exception extends directly or indirectly Exception class.

To handle checked as well as unchecked exception java provided 5 keywords.

try

catch

finally

throw

throws

**try and catch block**

syntax

try {

}catch(Exception e) {

}

Try block : the code which makes problem one line or multiple that we have to keep in try block.

Catch : This block execute only if any exception generate. Catch block is known as exception solver.

**ArithmeticException**

**Array topic**

In java array is known as reference data type.

Array is use to store more than one value of same types.

int a;

int b=10;

syntax

datatype arrayName[];

int []abc; declaration of array

int []xyz={10,20,30,40,50,60}

we have to retrieve the value using index position. Index position start from 0.

Scanner is a pre-defined class part of util package which help to take the value through keyboards.

**Syntax to create the memory size for the array and take the value through keyboards.**

datatype arrayName[]=new datatype[size];

int abc[]=new int[10];

float xyz[]=new float[20];

String name[]=new String[5];

Try with multiple catch block

**Finally block**

This block will execute 100% sure if any exception generate or not.

try

catch catch catch catch finally

catch finally catch

catch catch

finally

**finally block is use to close resources or clean up activity.**

**throw:** throw keyword is use to generate or raise the pre-defined or user-defined exception depending upon the conditions.

Syntax

throw new Exception();

or

throw new **ExceptionSubClass**();

**throws** : This keyword is use to throw checked or unchecked exception to caller methods.

Syntax

returnType methodName() throws Exception, ExceptionSubclass {

}

void dis1() throws Exception {

}

void dis2() throws ArithemticException {

}

ArithmeticException : divided by zero

ArrayIndexOutOfBoundsException : wrong index position

**Checked Exception**

We have to handle it.

Using try catch or throws mandatory

Checked Exception check twice at compile time as well as run time.

**IO Package**

**Input and Output Operation**

In Java we can achieve IO operation using **stream.**

**Stream:** Flow of data or it is abstraction between source and destination.

Input -🡪 Keyboards, File, Database, Networking etc.

Output 🡪 Console(Monitor), File, Database, Networking etc.

Stream

Byte Char

1 byte memory 2 byte memory

Input Output Input Output

InputStream OutputStream Reader Writer

They are type of abstract classes.

Byte wise classes

DataInputStream DataOutputStream

BufferedInputStream BufferedOutputStream

FileInputStream FileOutputStream

ObjectInputStream ObjectOutputStream

PrintStream

Char wise classes

InputStreamReader OutputStreamWriter

BufferedReader BufferedWriter

FileReader FileWriter

PrintWriter

System.out.println();

Scanner obj = new Scanner(System.in);

System is a pre-defined class part of lang package.

PrintStream ps = System.out;(console reference)

ps.println(“Welcome to Java”);

PrintStream always refers to standard output device.

Console or monitor.

InputStream is = System.in(keyboard reference)

InputStream reference always refer to standard input device ie keyword.

**Byte wise :**

**1st Example**

**Source :** keyboard

**Target :** Console

**2nd Example**

**Source : keyboard**

**Target : File**

**Buffer : It is a temporary which help to improve input as well as output operation.**

**1000 times improve performance.**

**File Buffer Program Buffer File**

**Char wise classes**

**Source : keyboard**

**Target : console**

1. Copy the file in upper case.
2. Copy each world first letter upper case.
3. Display the number of words present in the file.

**Object Serialization**

Storing object itself in external file or Storing object property in external file is know as object serialization.

Object -🡪 property (variables)

Behaviour (methods or functions)

Identity (reference names)

In object serialization we store only property.

Which class object want to serialized that class must be implements Serializable interface.

This interfaces doesn’t contains any methods.

**Java Bean class :**

**Day 2 : 11-03-2021**

**Object Serialization :**

Converting object into byte format or seriazable format or storing the object itself or storing object property into external file is known as object serialization.

Object De-Serialization : Converting byte format object into object format.

**Collection Framework : Data Structure**

int a;

int abc[];

Structure

class Employee {

int id;

String name;

Float salary;

}

Employee emp=new Employee();

emp.id=100;

emp.name=”Ravi”;

emp.salary=12000;

emp.salary=16000;

array object.

Syntax

Employee employees[]=new Employee[100];

int abc[]=new int[100];

in employees we can store only Employee class object. At beginning we have to plan how many records going to store. Remove, Search, find etc method very complex.

**Collection Framework (Data Structure) :**

Collection framework contains set of classes and interfaces which help to store the collection of elements or objects of any types (int, float, char, String as well as user defined class object).

All collection classes provide set of methods which help to any type of operation on those objects like add, search, iterate, remove etc.

Collection framework classes part of

util package

Collection Framework Hierarchy

Collection -🡪 interface

extends extends doesn’t extends

Set List Queue Map

All four are interfaces.

Set : It is use to store collection of elements or objects. Set doesn’t allow duplicate. Under Set Some API maintain the order or unorder or sorted.

List : It use to store the collection of elements or objects. List maintains the order using index. List allow duplicate elements or records.

Queue : First In First Out :

Map : It use to the information in the form key-value pairs. Key must be unique and value may be duplicate.

**Set :** Set implementation classes

**Doesn’t allow duplicate**

HashSet : Unorder

LinkedHashSet : Order

TreeSet : Sorted

**List :** List implementation classes

Allow duplicate

Stack : FILO

Internally extends Vector.

ArrayList : In Java ArrayList by nature it allow to store different type of values. Using arraylist we add or remote in between elements very easily.

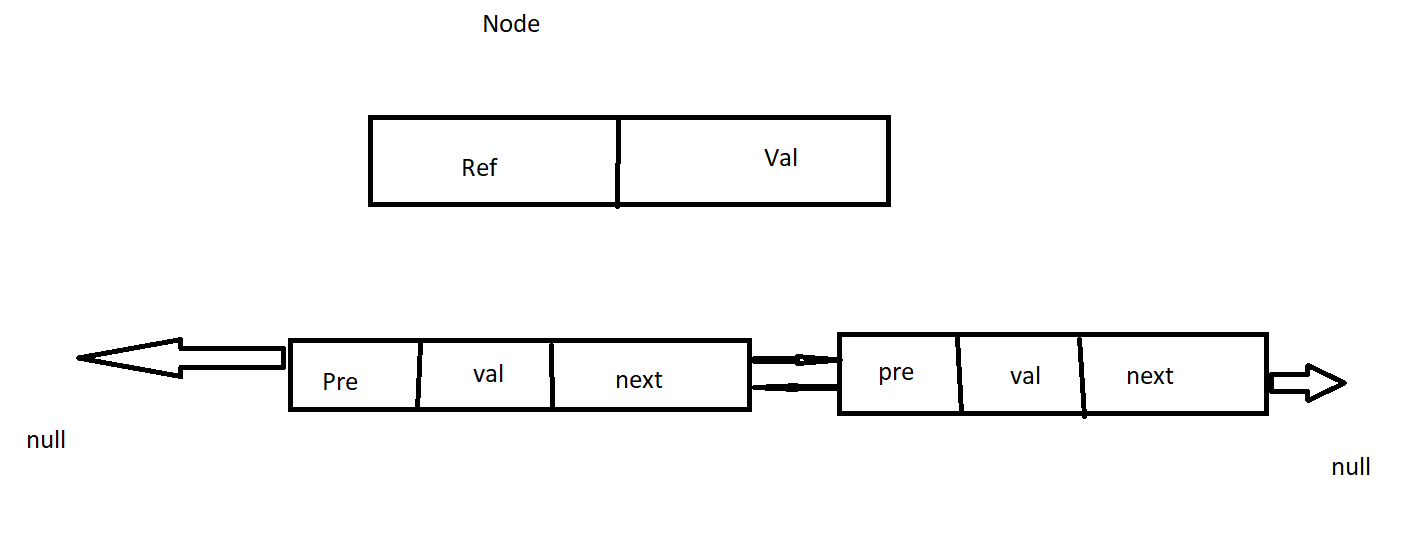
LinkedList : In Java By default LinkedList consider as double linked List.

Linked List always use Node to store the value.

Ref

REf

int abc[]={100,200,300,400,500};



ArrayList is good if we retrieve elements again and again using loop and Iterator.

LinkedList is good if we are doing more insertion or deletion operation.

**Vector :** It is a legacy. By default all method in vector are synchronized.

**Queue** : Queue implementation classes

**PriorityQueue** First in First out base upon the priority(by default take lower priority).

**Map :** Map implementation classes

HashMap : UnOrder

LinkedHashMap Order (stored Orded)

TreeMap Sorted Asc Order as a key

Hashtable By default method are Synchronized.

String str = new String(“Ravi”);

String str1 = “Ravi”;

Map: key – value pairs

Where key is unique and value may be duplicate.

Retrieve records from collection of classes.

**Set API**

Using for each loop

Using Iterator

**List API**

Using for each loop

Using Iterator : only forward direction

Using ListIterator : forward as well as backward direction.

**Collection Framework with Generics**

**Collection Framework with User-defined class objects.**

Whenever we display user defined class reference in println method internally it call toString method of Object class. That method return packageName.className@code

**VS Code**

**Node JS**

**Web Technologies**

https://[www.google.com](http://www.google.com)

req(http/https)--🡪

Client Server

🡨- res(http/https)----

**http:** Hyper text transfer protocol.

Set of rules which help to communicate more then machine or device.

**https:**

**www: world wide web**

**google : domain**

**com : commercial**

**URL : Uniform resource locator**

**HTML / HTML5:** Hypertext mark-up language.

It is use to display the content on browser in different formats.

**CSS or CSS3 : Cascading style sheet**

It is use to apply good look and feel for the contents or It is use to apply presentation for the contents.

**JS : JavaScript : JavaScript is use to provide action (events) on contents.**

HTML : HTML provide lot of pre-defined tags or elements.

HTML is a case insensitive.

Syntax

<tagName> opening tag

</tagName> closing tag

<tagName/> self closing

HTML Tags

1. HTML
2. Head
3. Body
4. Title tag
5. Paragraph tag : p
6. Heading tag : h1 to h6

H1 largest

H6 smallest

1. Br tag : break tag