Day 1

Agile

Git Overview

Java Basic Programming

Java OOPs Concept

Exception Handling

Packages

Lang, file handling, collection framework.

Data Structure using Java

Maven tool

Git Git is a distributed version control source code management tool.

SVN

Project : Java, .net, php, angular, react js

1%

10%

20%

A person Login module

B person Application module

C person Customer module

D person Employee module

SVN

1st Person push

Remote folder

2nd Person push

3rd Person push

Please download the git software

Install the software.

**git –version**

To make folder as a git repository we have to write the command as

**git init**

**git status :** This command is use to check the status of repository

git add filename : This command is use to send the file from file system to staging area.

git commit –m “some message”

This command use to pass the file from staging area to local repository.

**Git :** git is a open source distributed version control software tool which help to push or send the data to remote repository.

**git hub :**

AWS : code commit a to z.

Azure

Google cloud

git init

git add filename.txt

git commit –m “message”;

link local repository to remote repository

git remote add origin URL

**git push –u origin HEAD**

**Please create New folder**

**Open the git bash terminal**

**Git clone** [**https://github.com/Kaleakash/java\_oops\_phase1.git**](https://github.com/Kaleakash/java_oops_phase1.git)

**We will take the break…**

**Java : Java is a pure object oriented and platform independent programming language.**

**C : 1970**

**C with class or C++ : 1980**

**1990 :**

**Initial name of the Java is Oak :**

**Nov 1995 rename from oak to Java.**

**James gosling and Team**

**It was belong to sun micro system**

**Now it is a part of Oracle.**

**Version**

**1.0, 1.2……………..1.8,………………………………..16**

**Java 1.7**

**Java 8 Features 100%**

**Java 9**

**Java 11**

**Java 15**

**Java 8 Version**

**Simple Java program**

**Syntax of class**

**class ClassName { //**

**variables/fields;**

**methods/functions;**

**}**

**className must be follow Pascal Naming rules.**

1. **If class contains 1 world first letter upper case.**
2. **If class contains more than one world each world first letter uppers.**

**class Demo {**

**public static void main(String args[]) {**

**System.out.println("Welcome to Java...");**

**}**

**}**

**Save the program className.java**

**Demo.java**

**To compile the program open command prompt and refer to java program directory**

**javac Demo.java (javac className.java)**

**after compile successfully**

**java Demo (java classname)**

**Day 2 : 04-04-2021**

**Created 3 folder**

**Manager**

**In manager folder created Java class**

**Compiled and run the program**

**git init**

**git add .**

**git commit –m “java program created”**

git remote add origin https://github.com/Kaleakash/java\_projects.git

**HEAD last commit**

**git push –u origin master**

**git push –u origin main**

**git push –u origin HEAD (last commit in branch)**

**git push –u origin**

**origin means given the name for that URL (remote URL).**

**Ajay**

**git clone URL**

**given the instruction to Ajay write another java program with A.java**

**Vijay**

**git clone URL**

**given the instruction to Vijay write B.java program**

Branch : branch is just like a movable pointer which hold more than one commit details.

By default In git default branch created with name master/main

To check branch

git branch

Syntax to create the branch

git branch branchname

move from one branch to another branch

git checkout branchname

how to delete the branch

git branch –D branchName

1 way

Git branch branchname

Git checkout branchName

2 way

Git checkout –b branchName (created branch and switch to new branch)

Ajay Developer

Created branch using command

Git checkout –b Ajay

Created Ajay.java program

Compile and run

Git add .

Git commit –m “file created”

Git push –u origin HEAD

Then in remote branch (manager merge the Ajay from to main branch)

Then in Ajay developer come to main branch and delete ajay branch.

Pull in main/master branch from remote repository

Git pull ( from remote branch)

If you want to do any changes create the branch and push or delete if anything go wrong

Git push –u origin HEAD (From user – defined branch)

class Demo {

int a; //Ajay

int b; //Vijay

}

**Conflicts**

**Switch from one remote repository to another repository**

**Clone vs fetch command**

**Java Notes**

**class Demo {**

**public static void main(String args[]) {**

**System.out.println("Welcome to Java...");**

**}**

**}**

**Data types**

2 types

1. Primitive types : it is use to store only value

8 types

1. byte 1 byte
2. short 2
3. int 4
4. long 8 : without decimal
5. float 4
6. double 8 :with decimal
7. char 2 :single character
8. boolean 1 bit : true or false.

Data Type example

class Demo {

public static void main(String args[]) {

int a=10;

boolean b = true;

System.out.println(a);

System.out.println("The value of a "+a);

System.out.println("The value of b "+b);

}

}

Type casting

Converting from one data type to another data is known as type casting.

1. Implicit
2. Explicit

Int family

------------🡪 Implicit --------------------------🡪

byte short int long

🡨------------------Explicit ---------------------------------

class Demo {

public static void main(String args[]) {

// byte range -128 to 127

byte a=10;

short b=a; //implicit type casting

System.out.println(a);

System.out.println(b);

short c =129; //size 2

byte d = (byte)c; // (type)variableName; Explicit type casting

System.out.println(c);

System.out.println(d);

}

}

int to float family

by default every decimal number in java double consider. The memory size for double is 8 byte.

class Demo {

public static void main(String args[]) {

//implicit type casting

int a=10;

float b=a; //implicit type casting

System.out.println(a);

System.out.println(b);

//float c=10.10f;

float c = (float)10.10;

//double c = 10.10;

int d = (int)c; // explicit type casting

System.out.println(c);

System.out.println(d);

}

}

Byte casting char and int

class Demo {

public static void main(String args[]) {

int a=72;

char b=(char)a; // explicit type casting : char to int

System.out.println(a);

System.out.println(b);

char c ='A';

int d = c; //implicit type casting : int to char

System.out.println(c);

System.out.println(d);

}

}

Operator :

Arithmetic Operator : +, -, \*, /, %

Conditional Operator : >, >=,<, <=, ==, !=

Assignment operator : =

Logical operator : &&, ||, !

&& : both the condition must be true then only result true else false.

|| : if any condition true then result is true.

Increment and Decrement : ++, --

Ternary operator : condition ? true: false;

class Demo {

public static void main(String args[]) {

/\*int a=1;

int b=0;

int res = a&&b;\*/ // C/C++ but not in java

boolean a = true;

boolean b = false;

boolean res1 = a&&b;

boolean res2 = a||b;

System.out.println(res1);

System.out.println(res2);

}

}

Increment and decrement

Increment by 1

Decrement by 1

a++ : post increment

++a; : pre increment

If we are assigning the value to another variable or using inside a expression pre and post work differently.

++a: first increment and assign

a++ : first assign and increment

class Demo {

public static void main(String args[]) {

int a=10;

System.out.println(a);

a++; // increment by 1

System.out.println(a);

++a; // increment by 1

System.out.println(a);

}

}

class Demo {

public static void main(String args[]) {

/\*int a=10;

int b=a;

System.out.println(b);

b = a++; // increment by 1

System.out.println(b);

b = ++a; // increment by 1

System.out.println(b); \*/

int x = 3;

int y = ++x + x++; // 4 + 4

System.out.println(y);

System.out.println(x);

}

}

class Demo {

public static void main(String args[]) {

int a=10;

int b=50;

//int res =a>b?a:b;

String res = a>b?"a is big":"b is big";

System.out.println(res);

}

}

If statement

1. simple if
2. If else
3. Nested if
4. If else if
5. Switch statement

Looping

1. While loop
2. Do while loop
3. For loop
4. For each loop or enhanced loop

Loop is use to iterate or execute statement again and again till the condition become false.

Starting 3 types

Initialization start and end position

Condition true

Increment / decrement increment or decrement.

while loop

initialization

while(condition) {

body of the loop

increment / decrement

}

Do while loop

initialization

do{

body of the loop

increment / decrement

} while(condition)

For loop

1 2 4

for(initialization;condition;increment/decrement) {

3

}

class Demo {

public static void main(String args[]) {

/\*int i=1;

int n=10;

while(i<=n) { variableName

System.out.println(i +" , "+n);

i++; // n--;

}

System.out.println("Finish");

}\*/

/\*int i=1;

int n=10;

do {

System.out.println(i +" , "+n);

i++; // n--;

}while(i>=n);\*/

for(int i=0;i<=10;++i) {

System.out.println("i="+i);

}

}

}

While loop : we have to do the logic till the condition becomes false.

Do while : if we want to do one task without checking conditions like menu driven application

For loop : fixed iteration.

Continue and break

class Demo {

public static void main(String args[]) {

//break and condition.

for(int i=0;i<=10;i++) {

//if(i==5) break; // stop the loop

if(i==5) continue; // skip next line statement part of loop

System.out.println(i);

}

}

}

1. Non primitive types or Reference data types

It is use to store value as well as reference of another data types.

array :

class

interface

enum

array : it is user defined data types which is use to store the same type of values. It is also know as reference or non-primitive data types.

syntax

datatype arrayName[];

int abc[]; //java

int abc[10]; //C – C++ but not in java

array initialization

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

array values store using index position start with zero.

class Demo {

public static void main(String args[]) {

int a;

int abc[];

int mno[]={};

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

System.out.println(xyz.length);

System.out.println(mno.length);

//System.out.println(abc.length);

System.out.println(mno);

System.out.println(xyz);

//System.out.println(mno[0]);

System.out.println(xyz[0]);

}

}

For each loop or enhanced loop

Syntax

for(datatype variableName : arrayName) {

}

Display array using for loop and enhanced loop

class Demo {

public static void main(String args[]) {

int abc[]={100,200,300,400,500,600};

System.out.println(abc[0]);

System.out.println("using for loop");

for(int i=2;i<5;i=i+1) {

System.out.println(abc[i]);

}

System.out.println("Using for each loop or enhaced loop");

for(float a : abc) { // retrieve the element begining to end one by one

System.out.println(a);

}

}

}

Creating memory size for the array

Syntax

datatype arrayName[]=new dataType[size];

int abc[]=new int[10];

In Java new is a keyword which help to create the memory at run time (dynamic(.

Default values for array created using new keyword

Int family : 0

Float family : 0.0

Char : space (white space)

Boolean : false

String : null

Day 3 : 10-04-2021

Branch creation

Git branch branchName

Git checkout branchName

Or

Git checkout –b branchName

git clone URL

first time if you want to download the project from git repository we have to use

git clone URL

git clone URL

If you want to get the new updated data from existing git repository we have use git pull

Git pull : pull all files from repository and add the local repository.

git pull

Vs

git fetch : pull all files from repository but doesn’t add the local repository.

git merge

**Java OOPs Concept**

object : object is any real world entity.

Properties or State have variables / fields

Person

Behaviour do/does functions / methods.

Place

Accno, typeOfAccount

Bank

Withdraw, deposit, transferAmount

Animal

Wheel, name, price

Car   
 start(), appliedGear(), moving, stop()

Etc

Concept.

class : blue print of object or template of object or collection of objects which have same properties and behaviour or class is user defined data types which help to create the objet.

Syntax to create the object

ClassName objectReferenceName = new ClassName();