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)**