**Training Material**

What is GIT?

GIT VCS (version control system) - Used for storing files not just code also used for storing websites etc - and does version control.

The reason why we have VCS is for the following reasons:

* Collaboration
* Storing Versions
* Backup
* Improves visibility
* Accelerate product delivery

In VCS there are 2 Version control types, these are:

CVS - Centralised Version control system

DVS - Distributed version control system.

CVS definition

Diagram

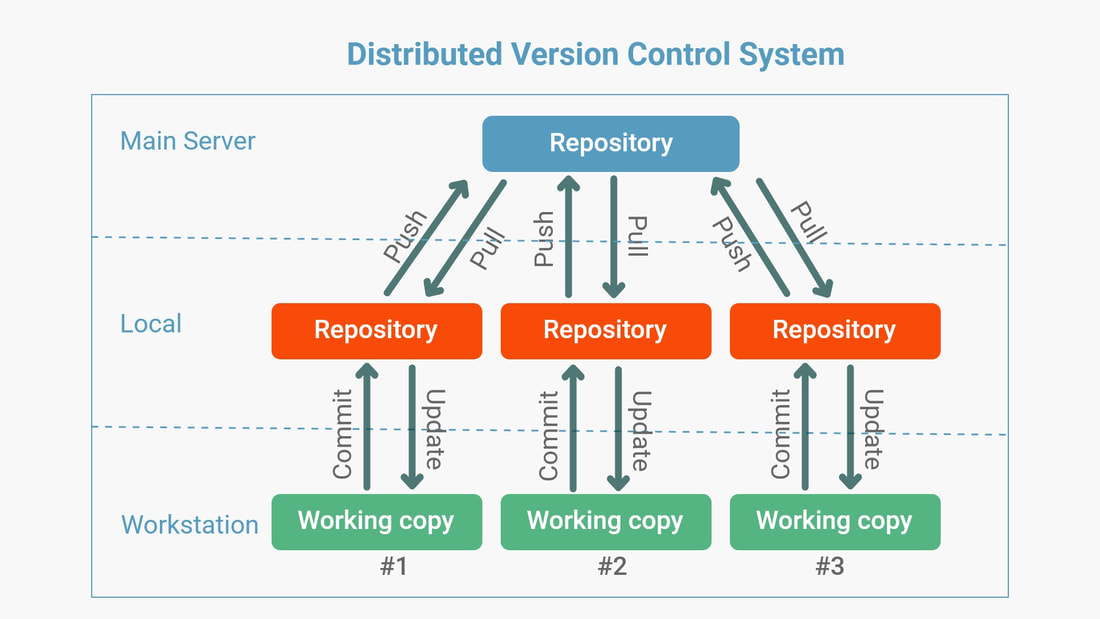
Description automatically generated

CVS is a repo on a centralised linux server/AWS.

For CVS, both people need to be online and if the CVS goes down it will mean you cannot work. - You both need to wait for the other person to push their changes to the repo.

An example of a CVS would be SVN.

DVS - Distributed version control system (GIT) definition



This has a central repo and branches off to a number of local repos which then branch off to working laptops. If centralised repo is down, then you are able to work on local copies. (The only time you will need to be online is when you are trying to push and pull from the main repo).

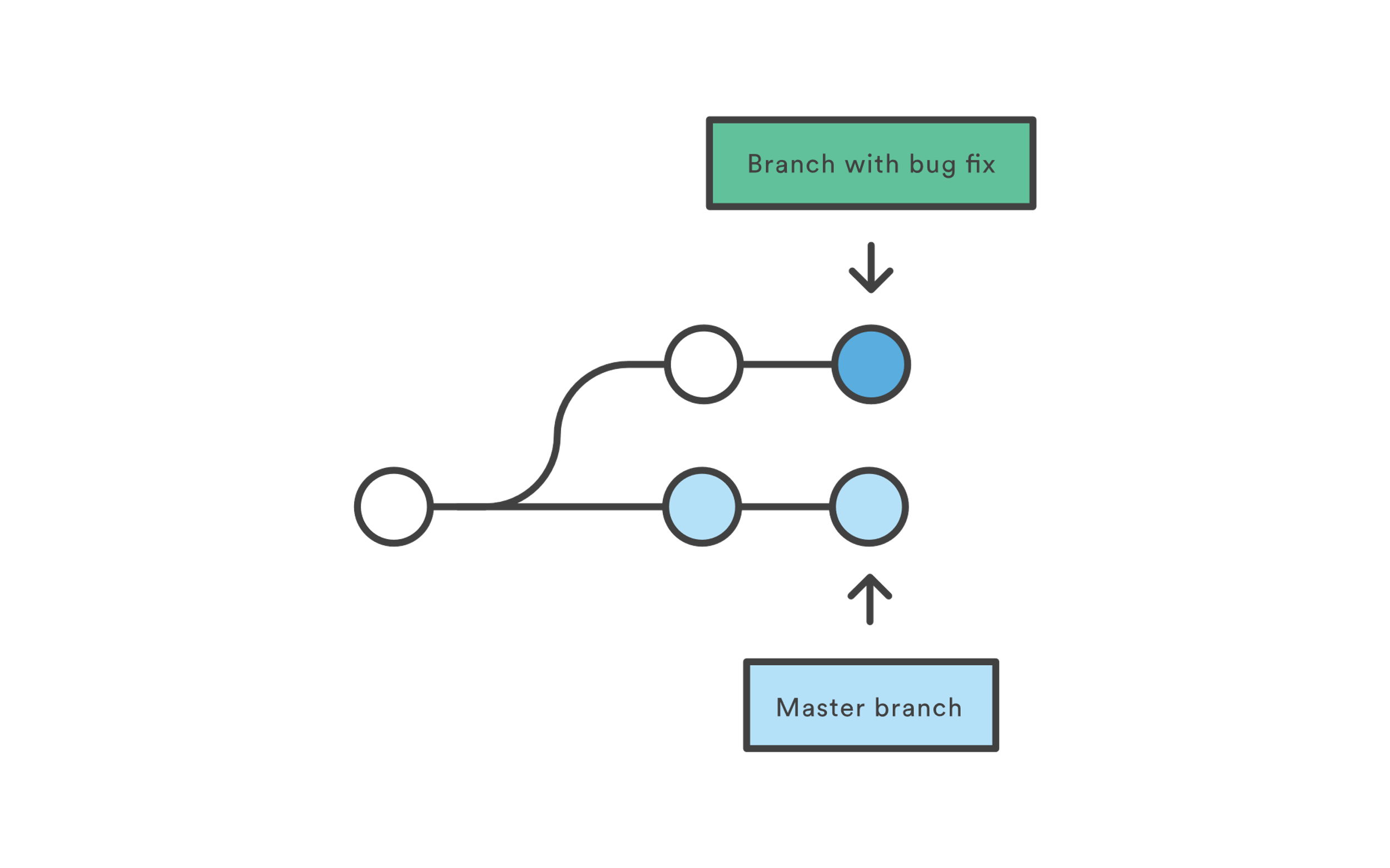
There is GIT (GIT is version control software) and Github/Bitbucket GUI (website) - They are not the same thing.

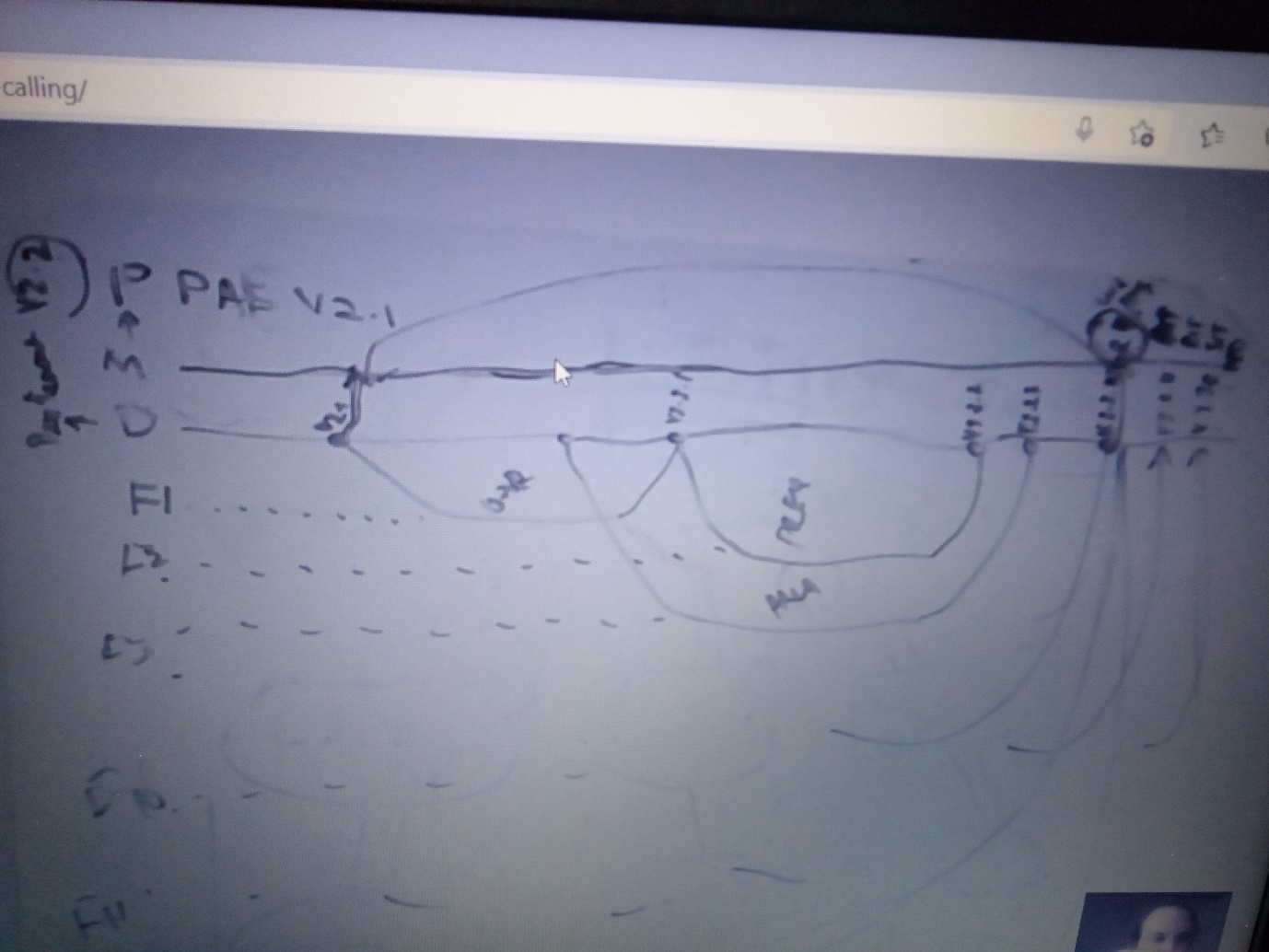
Any version control system can work with GitHub and opposite way round.

To download GIT you will need to go to the GIT website - you must check that its in the environment variables in the control panel in order to run GIT and check the GIT version.

GIT branching

As well as pulling down from the master repo, you can also branch off the master, this is done where if you wish to work a specific piece of work, an example is below:





GIT commands

In order to use GIT you need to ensure GIT bash is installed and use the following commands:

cd – Change Directory

git Status – Checks the status of the repository

git Init - This initialises the folder and creates a .git folder - this tracks changes made to git folder.

git add . - adds the file but does not commit need to do before committing the file.

git \* (add files specific)

touch \*.txt - This creates file in the directory

git commit -m "test" - This command commits the file to your local repo with a message about the file.

git remote add origin https://github.com/marktanner35/mark-demo2.git - This command initialises the folder in the main repository

git push -u origin master - This pushes the file into the main repository

git push -u origin main - This pushes the file into the main repository

git clone https://github.com/marktanner35/mark-demo2.git - This clones the main repo and creates a local copy

mkdir mark-demo2 - Makes a directory folder on your local

git add "filename" - This command adds the file to be committed if the file has been updated, you will then need to do a commit afterwards.

Creating a Git Repository

Before creating repositories, it is recommended to create an account in either Github or bitbucket. - In Github/bitbucket - You need to create a master repository and then do the following steps:

Create a repository, either by creating a repository in Windows or by right clicking and go to Git Bash here and use the command - mkdir then filename i.e. mkdir mark-demo2, you will then need to cd into the correct repository.

1. If you are intending on creating a file then you can use the command – touch then the filename – i.e. touch sample1.txt (this step can be skipped if this is not required).
2. Once this has been done you can then do a git status to check the status of your repository, this should then be followed by the command git init, this will initialise the repository and allow git to track version changes.
3. Once the repo has been initialised, you will then need to send a git add . to add the file, this will add the file to the repo before you will then need to commit the changes.

1. After you have added the file, you will then need to use the command git commit -m “” – this will commit the changes with a message i.e. git commit -m “test”.
2. Now on Github/bitbucket, you will need to create your repository, once you have done this, you can then retrieve the location of the repository and copy and paste this into GIT Bash – an example command would be: git remote add origin <https://github.com/marktanner35/mark-demo2.git>.
3. By using the command above, this will initialise the repo online in github/bitbucket, this means, we should now be able to “push” our changes to our repo in github/bitbucket. To do this, we need to use the following command: git push -u origin master.
4. Once you have done this, you will now see in github/bitbucket that your file is now there.

Cloning your git repo from github to your local

In order to clone your folder from github/bitbucket you will need to go to your repo in github/bitbucket and then click on “code” and use the address for “clone”, the following command then needs to be used in GIT bash:

git clone https://github.com/USERNAME/FOLDER.git - i.e. <https://github.com/marktanner35/test200.git>

**Training Material Lesson 2**

James Gosling invented Java working with Sun microsystems.

Java is platform independent, when writing a new tool so people think of making it compatible with Java – Open source is free – means you can get code and jar files. When you compile the code you will get a Jar file.

We add selenium Jar files to add to our automation framework.

Java is an object orientated program concept.

Object orientated programming

Object orientated programming is a programming style which uses classes, objects and other concepts such as inheritance, polymorphism, abstraction and encapsulation.

Class is a design which defines core properties and functions – i.e. A class could be a human being that contain body parts (data types) and functions such as walk, eat, sleep etc that make up that human being.

Inheritance – means inherit functions and properties from a main class, i.e. Male and female class would inherit core properties and functions from human being class.

Object – an instance of a class which has a physical existence – i.e. Human being named Mark is an object of Male class

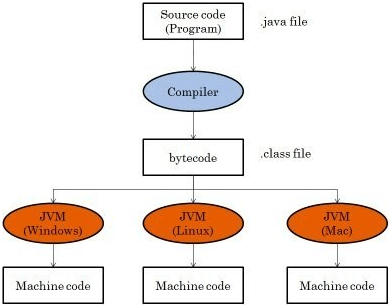
Abstraction – Means hiding information from the outside world – i.e. The mechanism of which we talk eat and sleep is hidden from outside world.

Encapsulation – Binding properties with functions – i.e. Hands can be used to hold things, legs can be used to walk etc.

Polymorphism – means taking “many forms” can be either overloading or overriding. Overriding example is if legs are used to walk normally what happens when change to walk backwards and overloading is instead of walking with legs, walk with hands.

How Java works

Sample.Java file ----- then we use a compiler (Javac contained in JDK) which converts to sample.class file (bytecode) ------ then JVM converts to machine code ----- then OS can view it.



Java program ---- compile ---- run (This run by Java runtime environment)

JDK – Java development kit – includes JRE and JVM

JRE – Java runtime environment – only has JVM

JVM – Java virtual Machine – Interpreter changes the class file to byte code which OS can understand.

Class files compressed and become Jar files.

Some programs require JDK as will need the specific components.

Download JDK then set the JAVA\_HOME add to the path and system variables – Please check TLG laptop for example on how to do this – Use Java 1.8

Download IntelliJ community version

IntelliJ Information

.idea – if change appearance of intelli J this is saved in .idea folder.

Pull request – from feature demo branch to master add comment and description

Find out what you can use in IntelliJ – Kikas recommendations:

If Menu disappears press ctrl+shift+a then search main menu then you will see the menu is switched “off” so toggle back “on”.

Use double shift to edit a configuration

Press “ctrl” twice to get the “run anything” box to run either your class or jar file configuration.

File – New – New project – give name, give location and package name name should be no space first letter capital of each name – make sure there are no spaces in between. Package is a lower case i.e. .com.qa.markdemo. Go to finish, you should create a project using template.

Once created you will have a main class (blue file) and main method in the class.

In IntelliJ – key words are highlighted in orange – i.e. Public static void main.

**Training Material Lesson 3**

Class is a collection of methods or functions.

Class will have a capital letter and main method will be in lower case.

A program needs a main method in order to execute a program i.e. if you want to run a jar file in a command prompt.

In order to create a package – right click src – new package – should be com.qa.packagename.

Iml file is used for tracking in Java like the .idea folder.

Class and file name need to be the same i.e if ever writing in notepad

“Public Class” – the public class is access modifier.

Public static void – “Static” you can call main method from anywhere. “Void” is what the method returns i.e. login screen

When declaring data types you take up Java memory.

// - represents comment for a single line

/\* represents comments which are blocks and more than one line.

For loops are used 80% of the time, while loops are also used.

Constants

Constants are values that do not change during program execution unlike variables which can change.

Java Operators

There are many Java Operators however really basic ones are:

Arithmetic Operators

* +
* -
* \*
* /
* % (Modulo – Returns remainder after diving 2 values)

Increment/Decrement Operators

* x++ (post increment)
* ++x (pre increment)
* x— (post decrement)
* --x (pre decrement)

Relational Operators

* > - Greater than
* < - Less than
* <= - Less than or equal to
* >= - Greater than or equal to
* == - equal to
* && - Means both statements are true
* || - Means one statement is true (is an OR statement)
* ! – Returns a false if result is true.

Primitive data types

* Double
* Float
* Char
* Byte
* Int
* Boolean
* Long
* Short

Reference data types

* String
* Array
* Class
* Enumeration
* Annotation

Writing a program (steps to think about)

1. Declare the variable
2. Write the operator
3. Then display the result i.e.System.out.println