Stage 1 - Programming Language - Java

Stage 2 - Selenium (Web Application) & AutoIT (Windows automation)

Stage 3 - Hybrid Framework

Stage 4 - CI/CD - Git and Jenkins

Selenium - <https://www.selenium.dev/>

* Open Source
* Language independent - Java, C#, Python, Ruby, Javascript, php, perl

Selenium - A suite of tools

1. Selenium IDE
   1. No need programming language
   2. Record and playback feature
   3. Supports - Chrome, Firefox, Edge
   4. Simple scripting or exploratory testing
2. Selenium RC - Depreciated
   1. Programming language is must
   2. Java, C#, Python, Ruby, Javascript, php, perl
   3. Architecture

Source code (Java+Selenium RC) 🡪 RC Server (Turn on/off) 🡪 Browser

1. Selenium WebDriver
   1. Programming language is must
   2. Java, C#, Python, Ruby, Javascript, php, perl
   3. Architecture

Source code (Java+Selenium WebDriver) 🡪 Browser

1. Selenium Grid
   1. Hub and Node

Java Programming -

1. Installation
   1. **JDK 8 (Java Development Kit)** 
      1. By default - it will install JRE (Java Runtime Environment)
   2. IDE
      1. Eclipse
         1. https://www.eclipse.org/downloads/packages/
      2. IntelIJ
      3. Netbean
2. Architecture

Source code (.java) 🡪 Byte code (.class) 🡪 O/P

Compile time vs Runtime

Compile time - Source code (.java) 🡪 Byte code (.class)

Runtime 🡪 Byte code (.class) 🡪 O/P

1. UpperCamelCase - MyFirstProject

lowerCamelCase - myFirstProject

1. Eclipse Structure

Workspace (lowercase)

Project 1 (UpperCamelCase)

Package 1 - lowercase (com.vfislk.purpose or org.vfislk.purpose)

Class (UpperCamelCase)

Methods & variable (lowerCamelCase)

Package 2

Project 2

1. Datatypes
   1. Primitive datatypes / Pre-defined
      1. Byte - 8 bit

7 bit for storing the number

1 bit for sign + or -

* 1. Non-Primitive datatypes / non-predefined - collection of primitive datatypes
     1. String
        1. Zero based index
     2. Array
        1. Zero based index
        2. Size is fixed

1. Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5

at com.vfislk.datatypes.DatatypesDemo2.main(DatatypesDemo2.java:17)

1. Relational operators -

==, !=, >,<,>=,<=

1. Logical operators - And (&&), or (II) not
2. Debugging
   1. Breakpoint
   2. Resume
   3. Terminate
   4. Step over
   5. Step into
3. Conditional statements
   1. If , if else, if else if
   2. Switch
4. Iterative statement
   1. For
      1. Start point
      2. End point
      3. Iteration
   2. For each/ advance for loop - designed for handling array and collections
   3. While
   4. Do while - minimum once it will run the code
5. Access modifier
   1. Private - accessible within the class
   2. Default - accessible within the package
   3. Protected - accessible within the package and also with inheritance
   4. Public - anywhere
6. Methods - Building block of the program
   1. Reusability
   2. Maintenance

* Static Methods
  + Create and Call it
  + To call the static method

Classname.methodname()

//accessmodifier static returntype methodname(arguments)

* Non-Static Methods
  + Create and Call it
  + To call the non-static method
    - Create object
    - Use objectref.methodname()

1. Object
   1. Declaration (Area obj)
   2. Instantiation - new (allocate memory)
   3. Initialization (Area())
2. Variable
   1. Static variable
   2. Non-static variable
3. Class & Object
   1. Real time example

Notes:

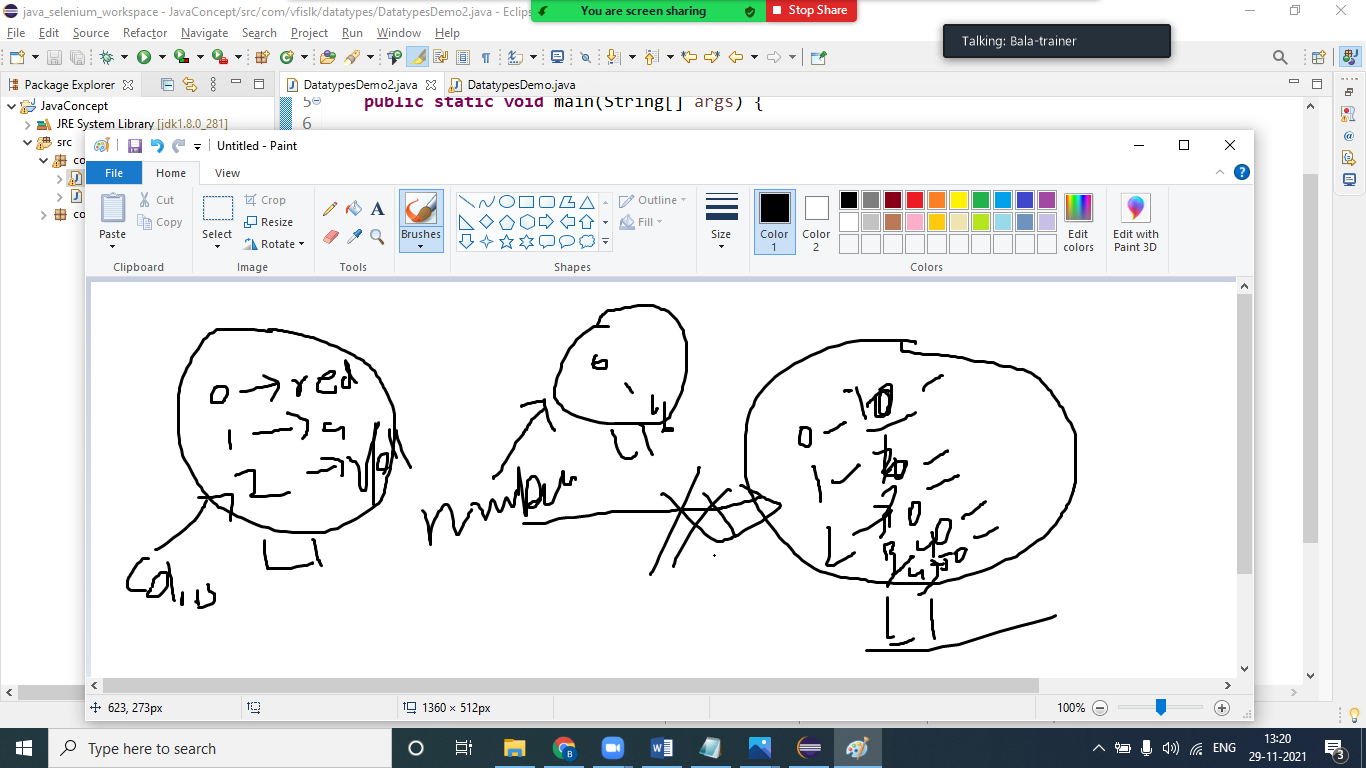
1 bit - 0 or 1

1B - 8 bit

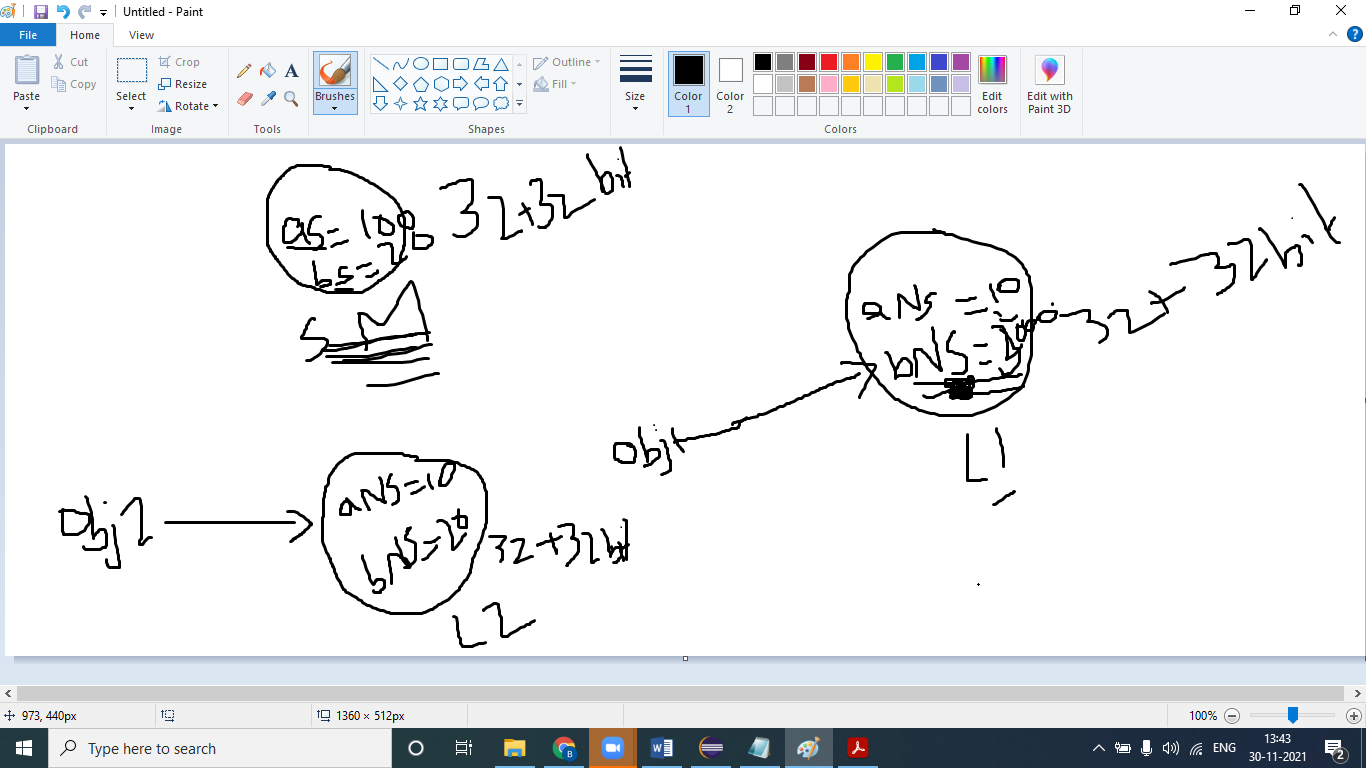
1KB - 1024B

1 MB - 1024KB

1 GB - 1024 MB



Ref 2



https://github.com/balaji-githubstore/VFISLK3.git

Task

|  |  |
| --- | --- |
| **PERCENTAGE** | **GRADE** |
| 90 and above | A |
| 80 to 89 | B |
| 60 to 79 | C |
| 45 to 59 | D |
| below 45 | F |

Task2:

1. Create Student type with below details (decide between static and non-static variable)

**Student**

* **studentRollno (public)**
* **studentName (public)**
* **studentMailid (public)**
* **studentPercentage (public)**
* **schoolName (public)**

1. **Create 3 different instance for storing below values**

**M1001,"jack",jack@gmail.com,45.2, Global school**

**M1002,"peter",peter@gmail.com,85.2, Global school**

**M1003,"mark",mark@gmail.com,56.5, Global school**