SDLC, Git, Shell Scripts

Scripts are nothing but commands stored in a proper order.

In DOS, script files are called as batch files.

1. Date
2. Time
3. Dir
4. Cls

Create a file called sample.bat

To create a batch file command is, “ copy con sample.bat “

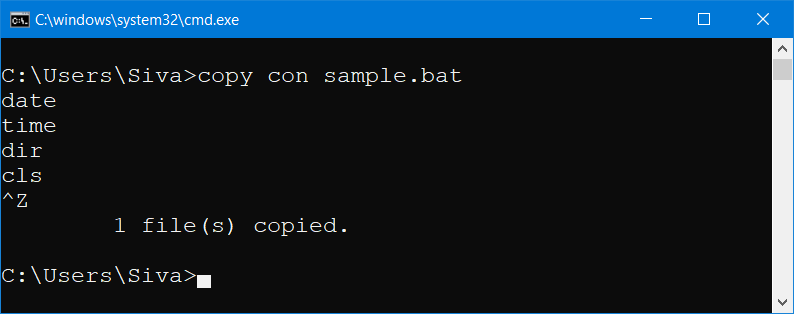
Date

Time

Dir

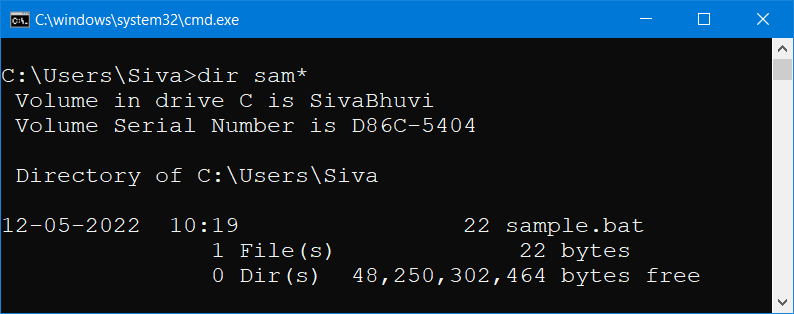
Cls

Press Ctrl+Z and Enter



To check the newly created file use the following command

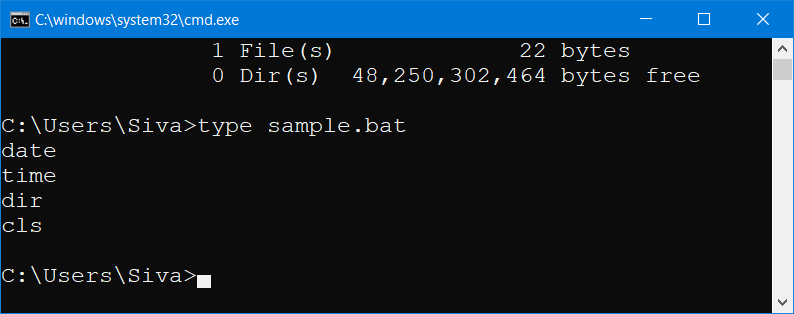
dir sam\*



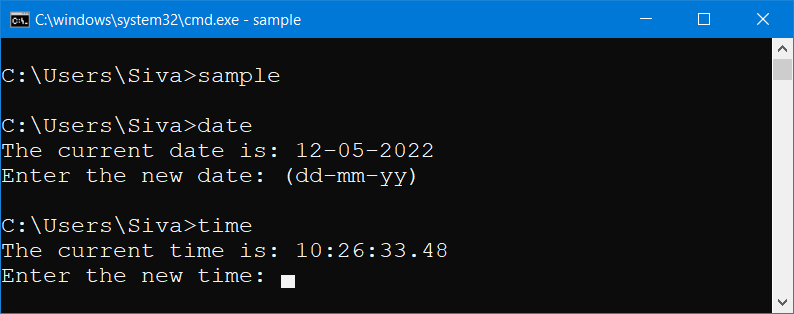
To check the file content of any files in dos, use the following command

Type <filename.ext> --- Syntax (A way to the use the command and all it’s allowed arguments)

Example type sample.bat

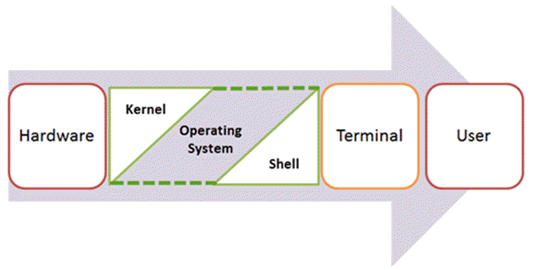


To run a batch file, just enter the name of the file without extension



In DOS, we can execute batch files (.bat), executable files (.exe) and command files (.cmd)

In Unix, we can execute shell script files (.sh)



In dos, the default editor is command prompt

In Unix,

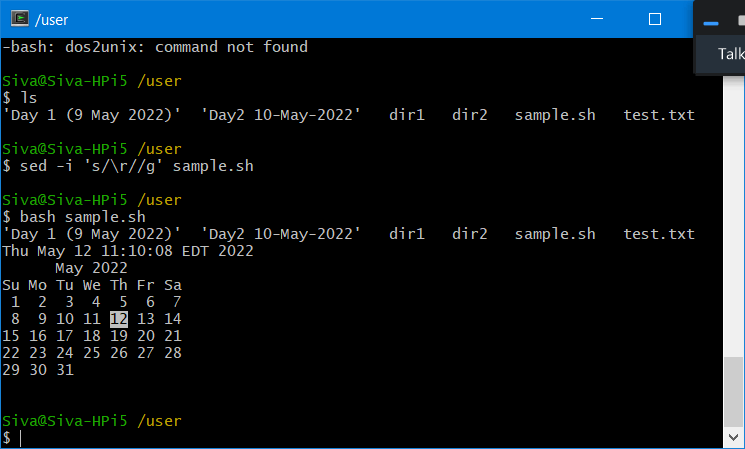
To create a file use touch or cat command.

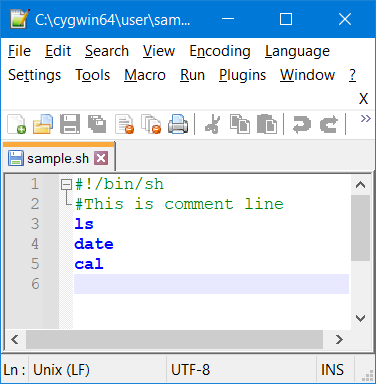
To read the file content use cat command along with the file name.

In unix, the default editor is vi editor

<http://www.gammon.com.au/vi> -- VI Editor informations

sed -i 's/\r//g' sample.sh [To Remove carriage return from the file sample.sh]

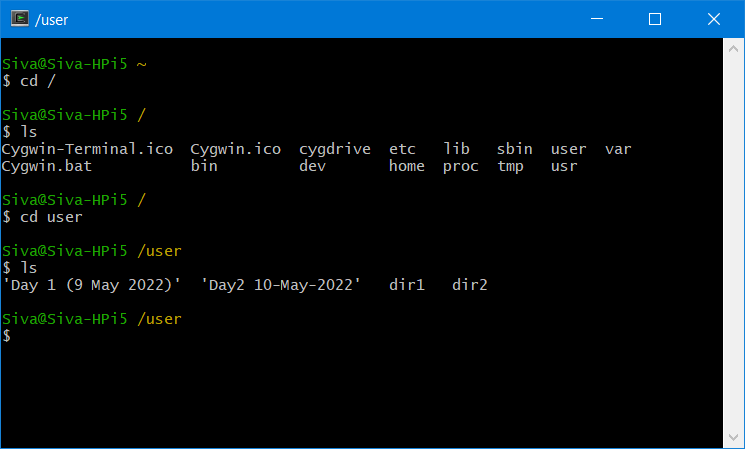




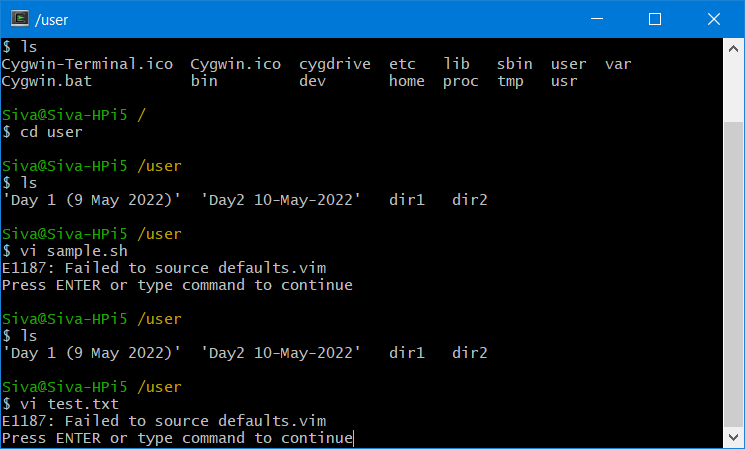
<https://askubuntu.com/questions/1148263/r-added-end-of-the-script-command>

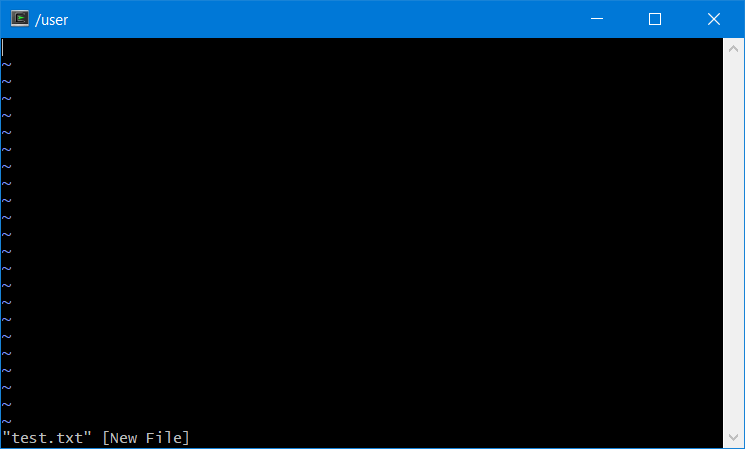
Creating a Shell Script

1. Open Cygwin



1. Create a new file using “vi” editor -- [vi sample.sh]





SDLC : Software Development Life Cycle

Phase in SDLC

1. Planning
2. Designing
3. Coding
4. Testing
5. Operations/Production
6. Maintenance

Earlier days, we used Water fall model. Now a days, we use agile (Sprint based)

Agile – Backlog (Product Backlog, Sprint Backlog, Burndown Chat)

To estimate the efforts we use T-Shirt sizing or Small, medium, large, Fibonacci numbers [ low, medium, high]

Epic, User Stories, Task

Daily Standup, Sprint Planning review, Spring retrospective -- Ceremonies in Agile

Scrum Master, Scrum member

Product Owner – The one who has given the requirement

Git – Version Control ( SCM – Source Code Management / Software Configuration Management)

Version controlling is a process of tracking changes made to the software source code.

Calculator.exe

1. Design the Front End
2. Design the BackEnd

Git is a Open Source Distributed Version control System which is created by Linus Torvalds (The one who created linux kernel)

Version Control Systems

1. SVN
2. Perforce
3. ClearCase
4. CVS
5. SubVersion
6. Git (Open Source) – unix like syntax – Git Bash

Git Bash – Is a command line client for Git (CUI) – Everything we will do using commands

Git GUI – Is a Desktop Client for Git (GUI) – Just click, drag & drop things will be done

JAVA – Java is a High Level, Object Oriented, Platform Independent, Multi-Threading, Secured Programming Lang.

High Level – syntax are almost similar to C

Object Oriented – Everything in Java is Object. Object is the root of the Language.

Platform Independent – Java Code can run in any OS with out re-compile. [WORA]

WORA – Write Once & Run Anywhere.

Java is a Platform Independent Language. But, JDK,JRE,JVM they all platform dependent.

You can download JDK & JRE from Official website.

SUN MicroSystem is the developers of JAVA.

Father of JAVA – James Gosling & Team

Oracle is taking care of Java and all future updates.

The latest version of java is Java18.

Java.com – We can download the JRE

JRE – Java Runtime Environment (To Run the Java Code/Byte Code .class file)

JDK – Java Development Kit (It contains Java Compiler and Other tools)

LTS – Long Term Support

Default Package of Java – java.lang (package name will be in lower case only)

Java is a Case sensitive Lang

Class name will always start with Capital Letter.

Variables & Method names will start with lower case only.

Constants will be represented using ALL CAPITAL LETTERS

JAVA uses Camel Casing – If it is more than One word, then First character of each word starting from the Second word will Start with capital letter only.

Valid Class Names (With Camel Case)

1. Employee
2. FamilyDoctor
3. MobilePhone

Valid variables & method Names

1. dateOfBirth
2. rateOfInterest
3. employeeId
4. emailAddress
5. mobileNumber
6. employeeName
7. getAddress();
8. findInterest();
9. displayResult();

Valid Constant Names

1. PI
2. NEW
3. MAX\_PRIORITY
4. MIN\_PRIORITY

Coding Conventions is followed globally by all Java Developers

Let’s assume we are all going to learn “German” lang [ Human Speaking Lang]

1. Learn the alphabets
2. Learn the words
3. Learn the Sentences, Grammar
4. Learn to write small paragraphs, stories, essays
5. Learn to write own stories, poems, essays, songs ….
6. We need a support language to learn new Lang (Mother Tongue / English) Reference Lang

We are all going to Learn JAVA [Computer Programming Lang]

1. Learn Alphabets [list of characters, list of numbers, list of symbols] [A-Z, a-z, 0-9, +,-,\*,/, %, @, .,,, “”,’’, $, |, &, #, ~, ]
2. Learn Keywords [ Pre-defined words] [All keywords will be in lower case] if, case, int, float, double, char, switch, for, while, do, class, interface, implements, extends, throw ….
3. Writing Expressions using variables & operators
4. Learn to write Class, methods, Objects etc.,
5. Learn Concepts of OOPs, Implementing OOP features
6. Adv Java Concepts

JAVA Editions

1. J2SE – Java 2 Standard Edition (Core Java – Used to create Stand-alone applications)
2. J2EE – Java 2 Enterprise Edition ( Adv Java – Web Based/ Enterprise Based applications)
3. J2ME – Java 2 Micro/Mobile Edition ( Small foot print of Java )
4. Platform Independent
5. Highly Secured
6. Multi-Threaded
7. No Pointer Manipulation (No explicit Pointer Manipulation)
8. No Multiple Inheritance (A class can extend only one class) [ Class Level]
9. Exception Handling
10. Supports OOP (Java is not a Pure OOP Lang)

Java is not a pure Object Oriented Programming Lang bcos of Primitive Data types

Android (Mobile Operating System) = JAVA + Linux Kernel

Solaris (Operating System) = Java Based

Big Data = Java Based

PEGA = Java Based (Code Less Application Development – BPM Tool)

Java is a General/Multipurpose Programming Lang. [Web/ Desktop/Mobile/Smart apps]

Java source code will have an extension called .java [HelloWorld.java == Java Source code file]

Java compiler will convert the source code into byte code (object code) [HelloWorld.class ]

HelloWorld.java (source code) -----javac (compiler)----🡪 HelloWorld.class (Byte code)

Q: Is a Java a compiled/interpreted lang?

Ans : Java is both compiled & interpreted programming Lang.

Exp: First the Java source code is compiled by the compiler which generates the byte code (.class file)

Then, the class file gets interpreted in JVM.

JVM = Java Virtual Machine (It’s a virtual Computer created by software codes only)

JVM interprets the byte code.

Compilation will convert all the lines to target format. (All the errors in different parts of the program will be reported)

Interpreter will convert one line at a time to the target format. (Error in a particular line will be reported and stops execution immediately)

Java Environment

1. JDK = Java Development Kit (Create/write Java programs, convert it into byte code. Generated documentation)
2. JRE = Java Run Time Environment

JDK is a SuperSet of JRE (JDK will also contain JRE)

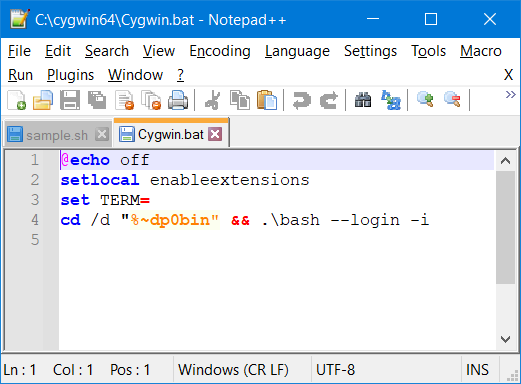
JRE is a sup set of JDK ( It will not contain all the tools of JDK)

IDE used for JAVA

1. Eclipse EE
2. NetBeans
3. IntelliJ IDEA
4. STS
5. VS Code
6. Atom

IDE – Integrated Development Environment

<https://docs.oracle.com/javase/8/docs/api/> -- JAVA 8 SE API Documentation



|  |  |  |
| --- | --- | --- |
| **Sl No** | **Angular** | **React** |
| 1 | Is based on TypeScript | Extended Java Script |
| 2 | Different types of Binding | One-Way Binding only |
| 3 | Normal DOM | Virtual DOM |
| 4 | It’s for Creating Responsive Web/Mobile apps | It’s only Creating Rich UI |
| 5 | Routing is supported by the framework | Will use 3rd party tool for routing |
| 6 | Used to Create SPA (Single Page Applications) | Used to create SPA |
| 7 | JS Based Framework | JS based Framework |
| 8 | Components are the building blocks of the framework | Components are the building blocks of the framework |

Computer Fundamentals

1. Architecture of Computer [ CPU, Input & Output Device]
2. CPU – Central Processing Unit which consist of processor, motherboard and different types of Memories
3. Input Device – Keyboard, Mouse (pointing input device)
4. Output Device – Monitor/Console, Printer [Dot-Matrix, Ink-Jet, Laser, Thermal Printers]

Types of Memories

1. RAM (Random Access Memory – Read & Write) – All the opened applications will be stored temporarily in RAM only [ 2GB to 32 GB ]
2. ROM (Read Only Memory) – BIOS, Boot Details will be stored [PROM, EPROM, EEPROM]
3. RAM & ROM belongs to Semi-conductor Memories
4. Magnetic Memory [Floppy Disk, Hard Disk]
5. Optical Memory [CD-Compact Disk (700MB), DVD – Digital Versatile Disk (4.9/9.4 GB), BRD-Blue Ray Disk – 47GB]
6. Chip based Memory [Pen Drives, Memory Cards etc., 64MB – 2TB]
7. Solid State Device Memory (SSD Memory)

OS Fundamentals

1. DOS & UNIX --- Character User Interface (CUI) based Operating System
2. Windows, Linux, Mac, Solaris, Chrome – Graphical User Interface (GUI) based Operating System

File System Management, Storage Management,

Multi-User, Multi – Tasking Operating System.

Java is a Strongly Typed Programming Lang. (In Java, we will define the data type of variables)

JavaScript is a weakly typed programming Lang. (It’s not compulsory to define the data type of variable)

Javascript variables are dynamically typed, meaning js variable can hold any type of values.

In JavaScript,

var x; //defining a variable called ‘x’

x= true; //Boolean type

x=35; //number (int)

x= 2463.874 //number (floating point)

x=’c’ //character

x=”welcome” //string

OOP = Object Oriented Programming [Class & Object]

We need to draw a circle. What are all the tools that we use to draw a circle??

1. Pen /Pencil
2. Marker
3. Paper
4. Compass
5. Coin of circular shape

Assume, we are going to draw the circle using coin.

1. Coin – Class [Blue print for Object]
2. Circle – Object [ Implementation of a class]

Everything in OOP is Object.

Object will have some properties & Behavior

Human – Name, DOB, Age, Email, Address, Father Name, Mother Name etc.,

Car – ModelName, YearOfMfg, NoOfSeats, noOfGears, engineType, transmissionType, speed

Properties are values in any form (numbers, String, etc.,)

Behaviors are changing the properties value

Accelerate() {

Speed++;

}

Decelerate/brake() {

Speed-=10;

}

Class – Is the blue-print for Object

Object – Instantiation of class (instance of a class)

Using a class, we can create n number of objects.

Shopping Mall (Design it based on the available space)

Drawing – Class (Blue-print)

In Java, we create the class using “class” keyword and we create object using the constructor of the class.

In Java, the file Name and Class name should be same. Java is Case Sensitive.

{ - Open Flower bracket or curly braces

} – Closing flower bracket or Curly braces.

Create a file named HelloWorld.java add the following code

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Welcome to JAVA");

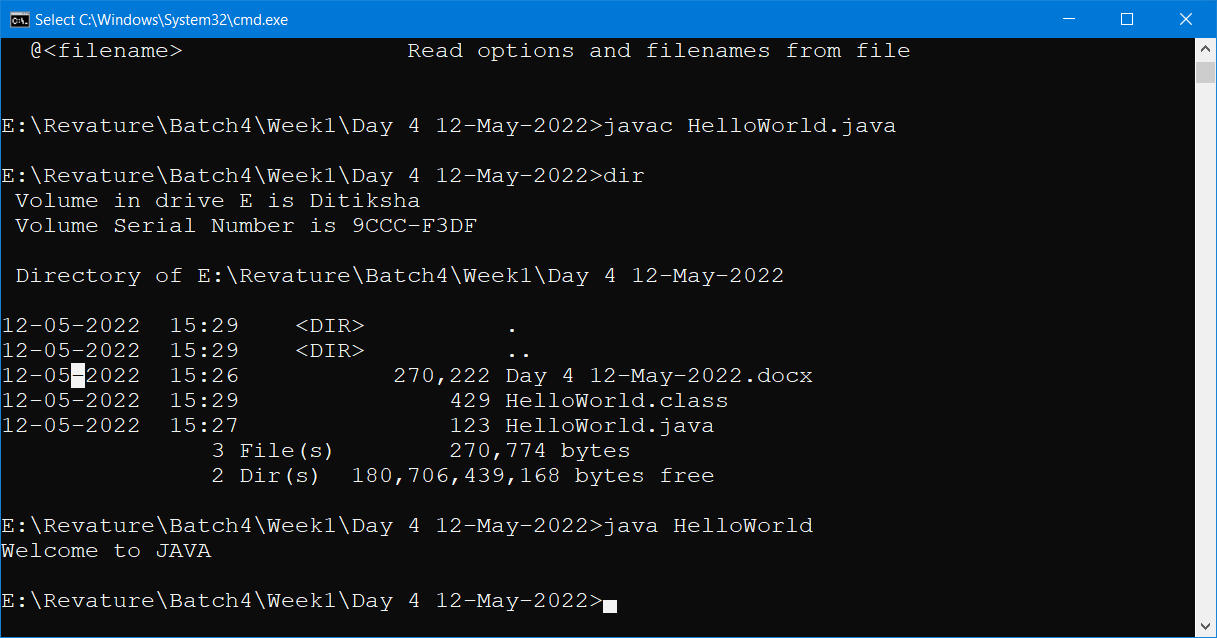
}

}

In windows,

Javac HelloWorld.java

Java HelloWorld



+ = Is a addition operator (when both operands are numbers otherwise it is a concatenation operator which combines string and other data type)

public class Calculate {

public static void main (String[] args) {

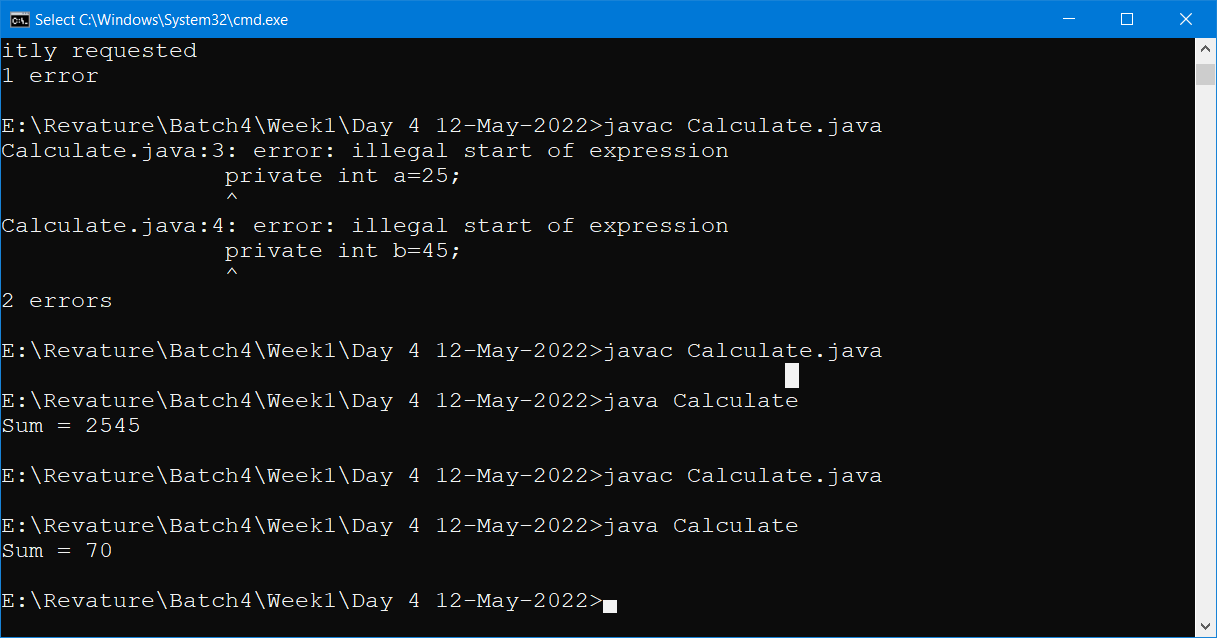
int a=25;

int b=45;

System.out.println("Sum = "+ (a+b));

}

}



**In Mac,**

javac --release 8 HelloWorld.java

Pillars of OOP

A = Abstraction ( Partial Class, Partial Methods ) – Hiding the implementation

P = Polymorphism ( Many forms, Method Overloading (Static) & Method Overriding (dynamic) )

I = Inheritance ( Inheriting the public and protected properties of Parent class )

E = Encapsulation ( Binding the Code & Data together – Hiding the Data )