DAY1

M/C lang- 0 1

Assembly lang: (LLL)------------Assembler---- 0,1

Store A,10

Store B,2

Add A,B

Low Level:---bYteCode--M/c lang

HL:-----into M/c lang

compile based-----------Compiler---m/c

interpretring lang.--------Interpreter-----------M/c

compiler ---whole code in one go

| Inetrpreter---10

2 Error--

5 Error

----------------------------------

Programming lang-compiler

Scripting Lang -Inetrpreter--supports OOPS

echo "Hello"

print "Hello"

document.write("");

non structural:BASIC, COBOL etc...

structural:C,PASCAL--Data Security

createAccount()

deposit()

withdraw()

------------------------------------------------------

oops:C++, JAVA

class - objects-- Security

Person---name, acountid,

banking

Classes and Objects

-----------------------------------------------------------

function/method:--

function a(){

//stmts

}

a();//calling

oops--methods

class A{

//method

public void m1(){

}

}

-----------------------------------------------

oops: class Object

class- template / blueprint---Map

objects- real world entity-Building-- instance of class

Animal-- Dog

Student (class)-- Rohan (object)

object having some prop and methods

class Student{

name; // variable //prop of an object

id

rn

//in the form of methods

public void eating(){

//

}

}

=============================================

OOPS:4 concepts

-----------------------

1. Inheritance:Parent- Child

Cal

Manual testing--> Automation--> API

Why use??---code Reusability

2. Encapsulation:- Data binding as a single entity.

Why use? data Security

How we can secure Data -private a=10;

3. Polymorphism:Many forms-

4.Abstraction:hide inner implementation

code complx. reduce

===================================================

jAVA---No full form

HLL or LLL ? ---HLL

compiled or interpreted?? - both

programming / Scripting - Programming

oops/ strcu. - OOPS

-------------------------

JDK-JAVA Devlopment Kit----->

JVM- Java Virtual Machine

JRE: + libs

-----------------------

GC--garbage collector

a=10;

10+20==M/m

a=10------> m/m -m/m address

a=10;

m1(){

}

a;

m1();

----------------------------------------------

1.Scource Code

First.java---fileName

class A{

psvm(){

}

}

2. compile code

Source Code -- -compile the code (JAVA compiler --- javac First.java)

Byte code---> LLL----> extension of Bytecode -.class

becoz of Byte code - we can sya Java is PLatform indep.

Wind10 OS--> compile--Byte code

Riun on any OS --MAC,Linux

3. Execution / Interpretation-- Interpreter (JVM)---java

Bytecode --> m/c lang--o/p

--------------------------------------------------

1. Source Code - Prog.java

file - - Test.class (Byte code)

class Test{

psvm{

//hello

}

}

2. compile the code

javac Prog.java <filename>//compiler generates byte code--(.class)

3. Execution of Byte code: JVM (Interpreter)

java Test <className>

hello

imp.

Is JAVA Platform independent?- PI

Is JVM Platform independent?- PD--window

JDK /JAVA-- window--JVM

JVM

DAY2

JVM ->JIT - Just in time compiler-- High performance

Source Code -.java--> bytecode -.class

->Byte code Verfier component- checks the byte code>JVM execute this Byte code

object--->prop to access the prop and Methods of class

object create-- new keyword-----M/m space allocate

object unused; GC

---------------------------------

Array:20 - 0-19

01 19

| 1|| |||||||

a =[1,2,3,4];

a[20]

-------------------------------------------------

illegal Data conversion: 1 byte- 8 bits--8 spaces

-------------------------

int a=10; //8421---1010

char ch = 'a';

double d = 12.34;

--------------------------------

byte b = 10; //1 byte ---8 spaces

short s=12;//2 bytes - 16 spaces

short sh = b;

byte bt =() sh;//not possible //compiler

10 lines

1

2--code; halt---Exception handling---Error

10/0---exception

try cATCH

3

4

5

6

7

THREAD....SMALL PROCESS,TASK

os-pROCESSSOR

t1-10 SEC--3SECS, T2 =5SECS,t3- 3SECS

class---code

FUNCTIONS1

func2

func3

---------------------------------------------------------------

Java Installation:

-------------------

DAY3

1. Source Code—anyname.java ---class Test
2. Compile code - - javac anyname.java --------------First.class
3. Execute the byte code – java First <classname>

class First <any classname>{

//psvm

public static void main(String[] args){

int a=10; // keyword indent = lit.

System.out.print("Hello JAVA");

}

}

Keyword,identifier, literal-value

class <keyword> ---small letters

First - <identifier--className>-- first letter should be in uppercase

className, variable name, method name

public –keyword /access – modifier

static-keyword

void- keyword

main-function name < identifier >

String -<builtin class Name> / identifier

[] – Array

Args-- <variable name / identifier >

String[] args : / String args[] / String … args

System….<class / identifier >

out—propertyname

print()—method name / identifier

predefined literals - values

null,true,false

reserved Words---Keywords , Literals

--------------------------------------------------------------------------------------------

Case1. Classname and Filename—Diff. – code execution done

Case 2. Classname and Filename—Same -- code execution done

Imp.Case 3. Public Classname and Filename—diff – get Error

Sol: Public Classname, save the file with the classname

Test.java

A.File

Public Class A{

Psvm (){

//

}

}

Class B{

Psvm (){

}

}

Javac Test.java—

Java A

Java B

Variables:

System.out.println("Hello");

System.out.print("JAVA"); //HelloJAVA

Var decl.—int a,b,c;

Var init.- int a=10;

# Datatypes:

# Primitive : can hold only single value in a variable:

Numeric : byte,short,int , long---10

Decimal: float - f , double

Char : ‘a’

Boolean: true / false---1 /0

**Non Primitive**: **: can hold multiple values in a variable**

String- “abc”

Arrays ---a = [12,23,45,67]

Classes and Objects