Aim of Worksop to build **strong programming skill**.

To solve logical program, you can select any programming language like C, C++, Java, C#, Python.

Majority programming language has certain common syntax and concept eg, Loop, If else, function. Recursive function, Array, String.

In this work shop we will use Java to write logical program.

Focus is on logic and mind development ,for that we will use java language. Focus is not on JAVA. You will learn this module separately.

I Know only basic of ‘C’ how should I cope.

Observe few differences in syntax.

|  |  |
| --- | --- |
| C | Java |
| Procedural programming language | Object oriented programming language |
| Eg. Hello world in C | Eg. Hello world in java |
| #include<stdio.h>  int main(int argc, char \*argv)  {  printf("Hello world");  return 0;  } | **import** java.lang.\*;//importing classes/ methods from package  **public** **class** Myclass **{**  //public Access modifier  **public** **static** **void** main(String[] ar) {  //System is final class  //out is static member  //out is of type printstream    System.out.print("Hello world");  }    **}**  **//save your file with class name**  **Myclass.java**  **After compilation Myclass.class** |
| #include<stdio.h>  printf() is a function who’s prototype is in this header file | **import** java.lang.\*;  System is a class which is available  In java.lang.\*; ie. A folder having all other classes. |
| int main()  entry point for application | **public** **static** **void** main(String[] args) {  entry point for application |
| It is function based procedural programming language | **public** **class** Myclass {  }  Since it is object oriented every thing are encapsulated ie. Wrapped in a class |
| After compilation it will generate EXE file  Extension of a file is .c | After compilation it will generate .class file  Extension of a file is .java |
| Can give any name to the file | File name and class name has to be same. |
| Printf is easy to write | You can shorten this line  System.***out***.print to ***out***.print by adding this line  **import** **static** java.lang.System.***out***;  **Static import** is a feature introduced in the **Java** programming language that allows members (fields and methods) which have been scoped within their container class as public **static** , to be used in **Java** code without specifying the class in which the field has been defined. |
| Os will call exe file and can directly affect OS | Java virtual machine will call .class file and it run in java runtime environment |
| How to take input  scanf(“%d”,&no);  object  reference  4000  Public nextIint nextInt()  Heap  5  no1  Stack  4000  Scanner  2000  sc  2000 | How to take input  **import** java.util.\*;  Scanner sc= **new** Scanner(System.in); //System.in is a standard input stream.    out.println("Enter a number");  no1=sc.nextInt();  for float 🡪 sc.nextFloat()  Double -🡪sc.nextDouble()  Byte🡪sc.Byte()  String🡪sc.next() //string  **char** c = sc.next().charAt(0);  //Anita🡺’A’  // 65+32🡺97 |

**Structure of System Out Println in Java**

The statement System.out.println contains three major parts: System, out, and println.

* **System** refers to a final class that we can find in the java.lang.package and this System class has a public and static member field called PrintStream.
* All instances of the class PrintStream have a public method called **println()**
* **out** is an instance of the PrintStream class

Public final class System

{

static PrintStream out=new PrintStream.();

}

System.out.println(“hello”)

Public class PrintStream

{

Public int println(String s)

{

}

}

PrintStream

Public int println(String s)

out

byte: The byte data type is an 8-bit signed two's complement integer. It has a minimum value of -128 and a maximum value of 127 (inclusive). The byte data type can be useful for saving memory in large arrays, where the memory savings actually matters.