4) Punctuators :
---
* It is also known as Separators.
* It describes to the compiler that how the things will be grouped together.
  Example : , ; ... (var args) () {} and so on

What is a local variable ?
---
* If we declare a variable inside the **method** OR **block** OR **constructor** body then it is called Local /Stack/ temporary/Automatic Variable.

  Example :
  public void accept()
  {
     int x = 100; //local variable
  }

* A local variable must be initialized by the developer before use.

* A local variable must be pre-declared and initialized before use.

  public void accept()
  {
     System.out.println(x); //error
     int x = 100;
  }

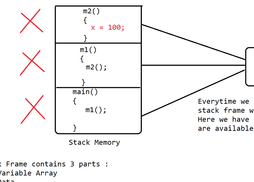
* **We cannot apply any kind of modifier on local variable except final;**

* As far as it's accessibility is concerned, It is accessible within the same method body OR block OR constructor that means we cannot use/access local variable outside of the method body OR block OR Constructor.

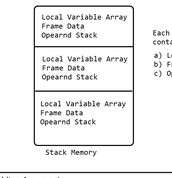
Q) Why we cannot access local variable outside of the method/block/constructor body ?
---
* All the methods are executed in a very special memory called **Stack Memory** which works on LIFO basis.
* In java, Whenver we call a method then a separate STACK FRAME will be created for each and every method.
* This STACK FRAME is responsible to hold the value of local and parameter variable.

  //Program
```
package com.ravi.method_execution;

public class MethodExecutionFlow
{
    public static void main(String[] args)
    {
        System.out.println("Main method started!!!");
        m1();
        System.out.println("Main method ended!!!");
    }

    public static void m1()
    {
        System.out.println("M1 method started!!!");
        m2();
        System.out.println("m1 method ended!!!");
    }

    public static void m2()
    {
        System.out.println("m2 method stated!!!");
        int x = 100;
        System.out.println(x);
    }
}
```

 Diagram for the above Program :



* Every Stack Frame contains 3 parts :
  a) Local Variable Array
  b) Frame Data
  c) Opearnd Stack



Limitation of Command line Argument :

As we know by using Command Line Argument, we can pass some value at runtime, These values are stroed in String array variable and then only the exceution of the program will be started.
In Command line Argumenet we can't ask to enter the value from our end user as shown in the Program.

//Program that describe we cannot ask for user friendly input message using Command Line Arg.
```
package com.ravi.command_line_arg;

//Reading a character by using Command Line Argument
public class LimitationOfCommandLineArgument
{
    public static void main(String[] args)
    {
        System.out.println("Enter your Gender [M/F]");

        char gender = args[0].charAt(0);
        System.out.println("Your Gender is :"+gender);
    }
}
```

public char charAt(int indexPosition) :
---
* It is a predefined non static method of String class.
* It is used to retrieve a character from the given String based on the index position.
* The return type of this method is char data type.

Reading the Data from the Client by using User friendly message :

We can read the data from the Clinet with user friendly input message by using following :

 1) java.io.DataInputStream class
 2) java.io.BufferedReader class
 3) java.lang.System.in.read();
 4) Using java.io.Console class [Used to read password]
 5) java.util.Scanner class

Scanner class :
---
* It is a predefined class in java.util package from JDK 1.5V.
* Using Scanner class we can read proper input with user friendly input message.
* It contains various non static methods to read the appropriate data from the clinet.

Static variable of System class :

System class has provided 3 final and static variables which are as follows :

 1) System.out : It is used to print normal message on the console
 2) System.err : It is used to print error message on the console.
 3) System.in   : It is used to take input from the sourse.

How to create an Object for Scanner class :
---
* In order to create the object for Scanner class we should use new keyword :

```
Scanner sc = new Scanner(System.in);
```

Non static methods of Scanner class :
---
1) public String next() : Used to read a single word.
2) public String nextLine() : Used to read multiple words OR a complete line
3) public byte nextByte() : Used to read byte value
4) public short nextShort() : Used to read short value
5) public int nextInt() : Used to read int value
6) public long nextLong() : Used to read long value
7) public float nextFloat() : Used to read float value
8) public double nextDouble() : Used to read double value
9) public boolean nextBoolean() : Used to read boolean value
10) public char next().charAt(0) : Used to read char value.