Variables in Java:

Instance Variables :

1. Such variables which are directly created within a class outside any block or method.
2. For every object separated copy of instance variables will be created.
3. Memory for these variables would be given on the heap area, so jvm would perform initialisation with default values.
4. Variables whose value changes from Object to object (i.e for every new object new memory map on heap and new values it will hold)

```
public class Student{
    private String sname;
    private int sid;                /// Instance varibales
    private int sage;

        public Student(String sname,int sid,int sage){
                this.sname=sname; this.sid=sid; this.sage=sage;
        }
}
```

Note : If we make instance variable as final, then compulsorily we should perform initialisation explicitly jvm wont provide default values. (bcz final variables act constant - watch the class lecture on final keyword)

Whether we use or not otherwise we get compile time error.

Local variables :

1. Variables which are a part of method signature, method body and those variables are called as "Local variables". (for the first time we declare within a method or any block)
2. Variables which are created by the programmer to meet the temporary requirements are called "local variables".
3. Memory for those variables will be given on the stack memory,jvm will not initialise any value for those variables, compulsorily the user should initialise the value.

```
eg:: class Test{
        public static void main(String... args){
                int i;  // local variable
                System.out.println("Hello");
    }
}
```

Note : If the local variables are final then we should initialise before we use.(final keyword topic done during Inheritance concept kindly watch)

The only modifier applicable for local variables is final,if we use any other modifier it would result in a compile time error.