Session 1 : 22-Jan-20

**To set compiler and interpreter path for java**

Go To start -> right click on Computer -> select Properties

Select "Advance System Settings" option from left side of the window

Select "Environment Variable" button

Select "New" button from "user variable for java" section

Enter Variable Name as "path"

variable value : "C:\Program Files\Java\jdk1.8.0\_25\bin"

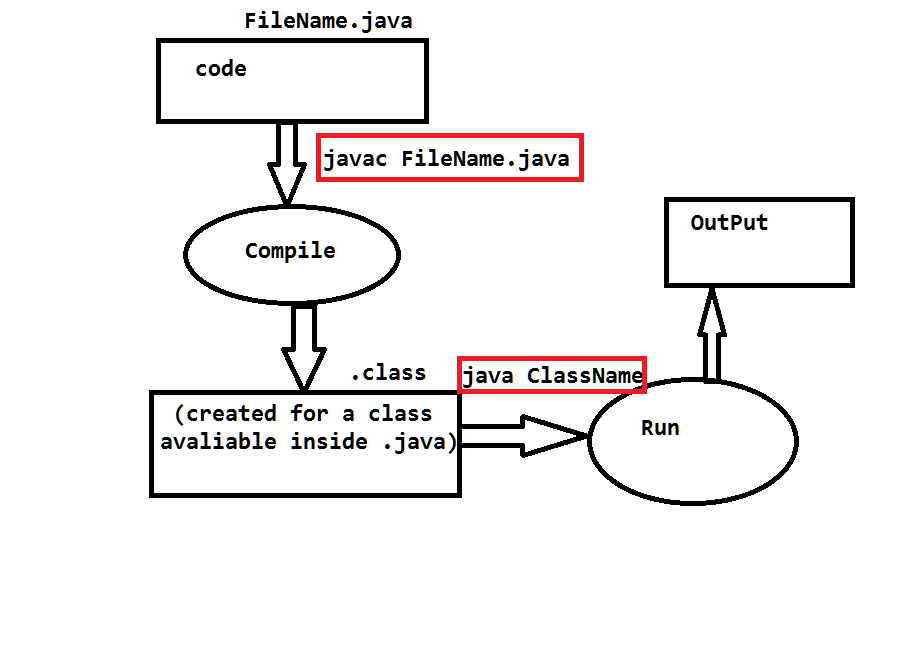
**To Run Java Program**

1. Compile a program

**javac fileName.java**

1. To Run Proram

**java className**



1. Java file name must be same as public class name.
2. To execute java program we required main method,

If java class do not have main method then we can compile java

Class but get an error at execution time.

1. one java file can have more than one classes but can create only one

Public class.

1. .class files gets created for classess present inside java file

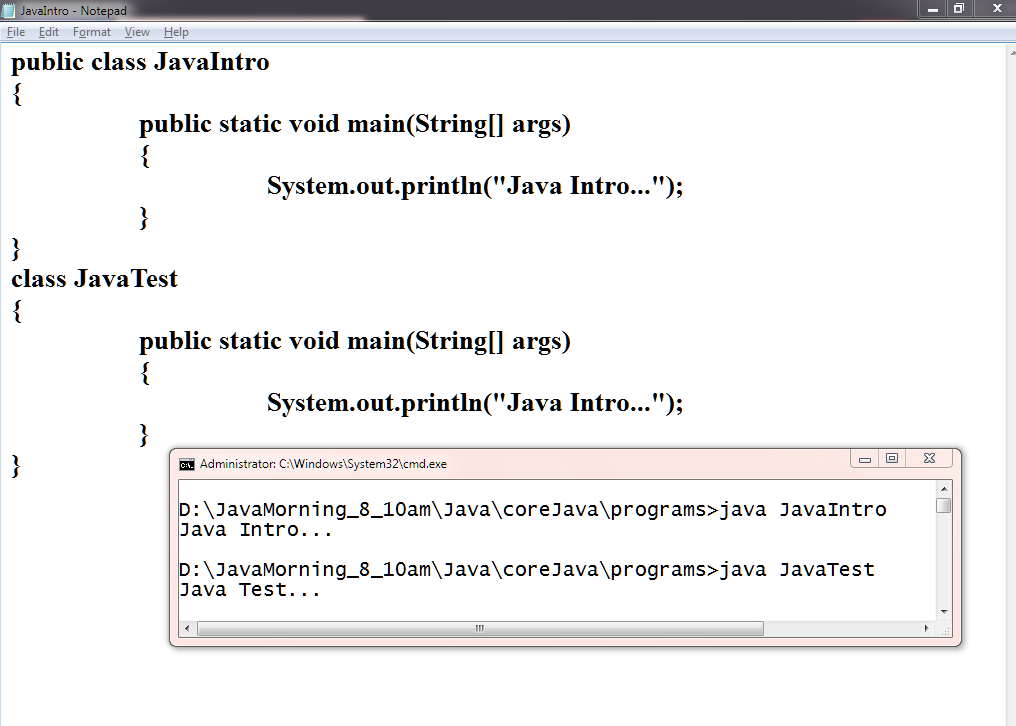
If we create 2 classes inside single java file then after compilation we

Will get 2 .class file for each class.

1. One java file can have more than one main method in different

Class. And each class can execute independently.

1. Java file can have more than one class but only one class can be public, we cannot create more than one public class in a java file



Session 1 : 23-Jan-20

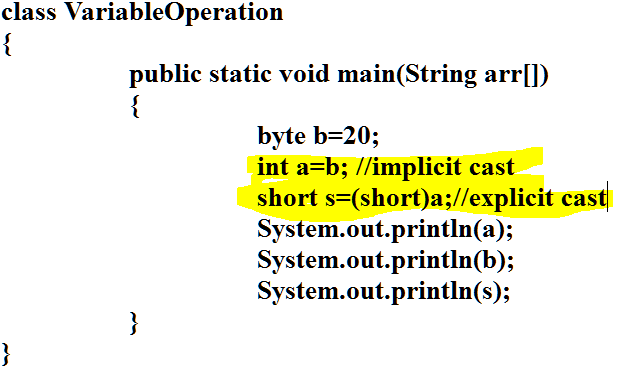
**Data Type**

1. Premitive data typ -> These are the data type which is having fix size
2. Non-premitive data type -> Not having fix size

|  |  |  |
| --- | --- | --- |
| **Type** | **Premitive Data Type** | **Size** |
| Number | byte | 1 byte |
| short | 2 byte |
| int | 4 byte |
| long (use 'L ' or 'l' as a suffix of long value-not mendetory) | 8 byte |
| Float | float (use 'F ' or 'f' as a suffix of float value-mendetory) | 4 byte |
| double (use 'd ' or 'D' as a suffix of double value-not mendetory) | 8 byte |
| text | char(unsigned int)(rang : 0-65535 ) | 2 byte |
| logical | boolean | 1 bit |

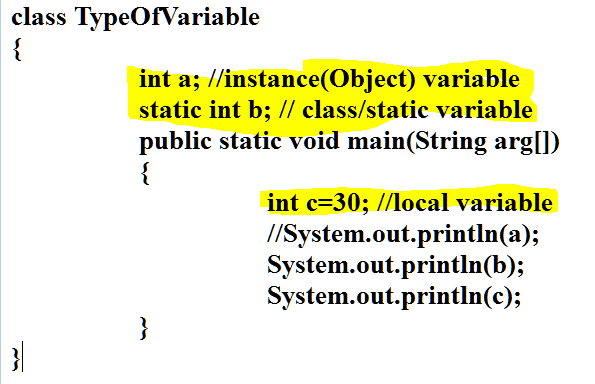
**Preimitive Data Type casting**

1. **Implicit cast** : will be done automatically by Java with no risk
2. **Excplicit cast** : have to do it manually by developer and its own risk



**Type of Variable(Scope Of variable)**

1. **Instance variable**
   1. The variable declare inside class and outside any method
   2. All instance variable gets initialize by default values if user not provide any value
2. Static/Class Variable
   1. The variable which is declare inside class and outside any method using static keyword
   2. All static variable gets initialize by default values if user not provide any value
3. Local variable
   1. The variable which is declare inside method or constructor
   2. All local variable has to initialize before use.



**Default Values for instance and static or class variable**

|  |  |
| --- | --- |
| **Data Type** | **Default Value** |
| byte  short  int  long | 0 |
| float | 0.0f |
| double | 0.0d |
| char | \u0000 or 0 |
| boolean | false |
| All non-primitive data type | null |

Session 3-4 (27-28-Jan-2020)

**Control Flow statement**

1. Sequecnce
2. Conditional
   1. If statements
   2. Switch cases
3. Looping
   1. While
   2. Do-while
   3. For
   4. Enhance for(for each)

**while loop: While loop perform pre checking**

For while loop has to provide 4 things

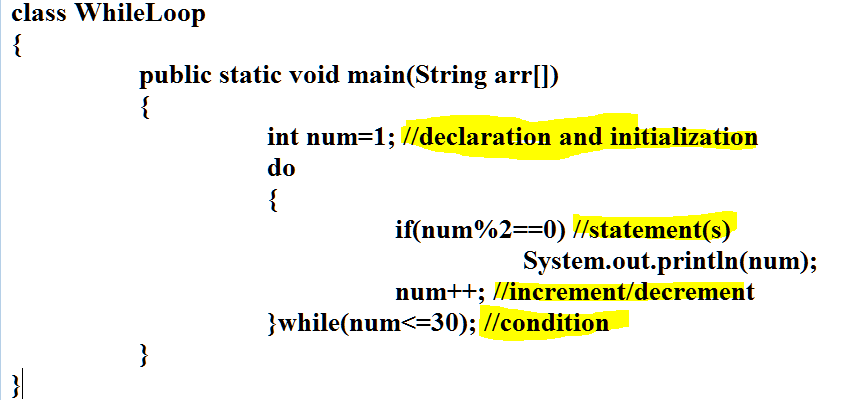
1. Declaration or inititalization of variable
2. Condition
3. Statement(s)
4. Increment/Decrement

**do-while loop: performs post checking**

For while loop has to provide 4 things

1. Declaration or inititalization of variable
2. Statement(s)
3. Increment/Decrement
4. Condition

While loop perform pre checking



**for Loop :**

For while loop has to provide 4 things

1. Declaration or inititalization of variable
2. Condition
3. Statement(s)
4. Increment/Decrement

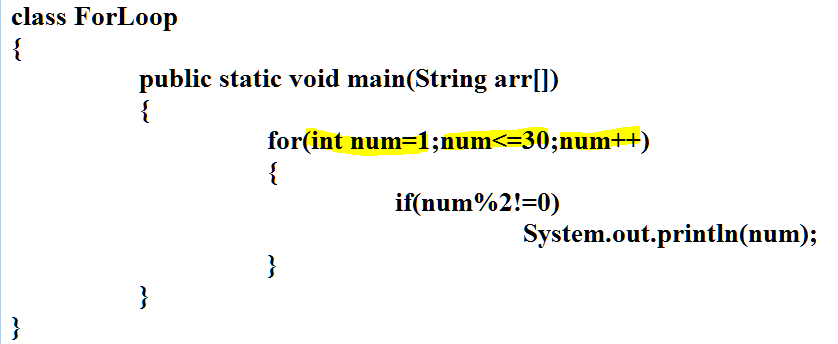
Syntax

for(declaration/initialization **;** condition **;** increment/decrement and statement(s))

{

Statement(s)

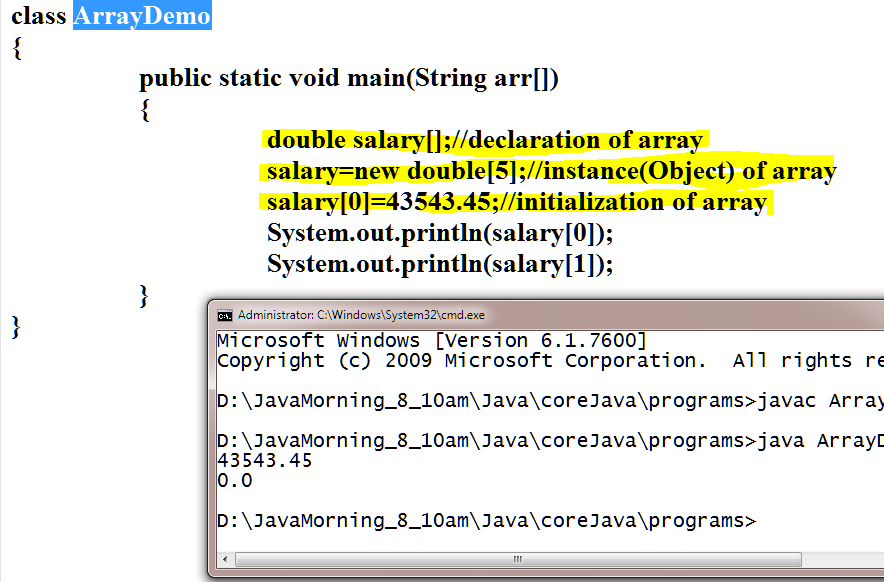
}

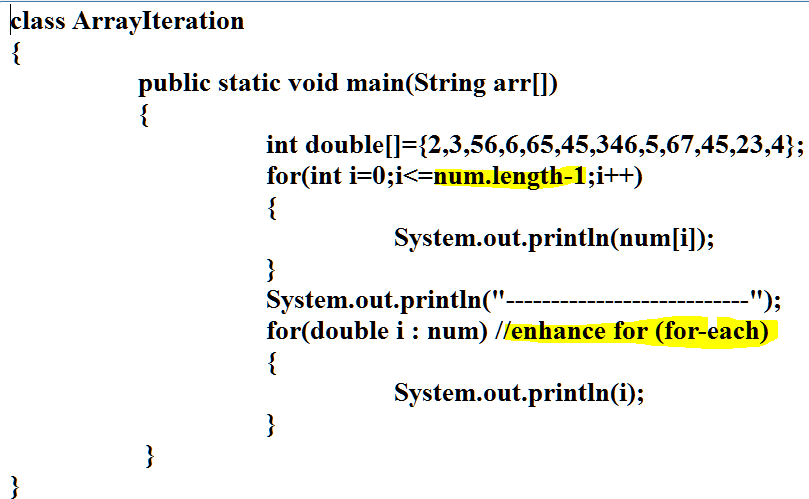


Session 29-jan-2020

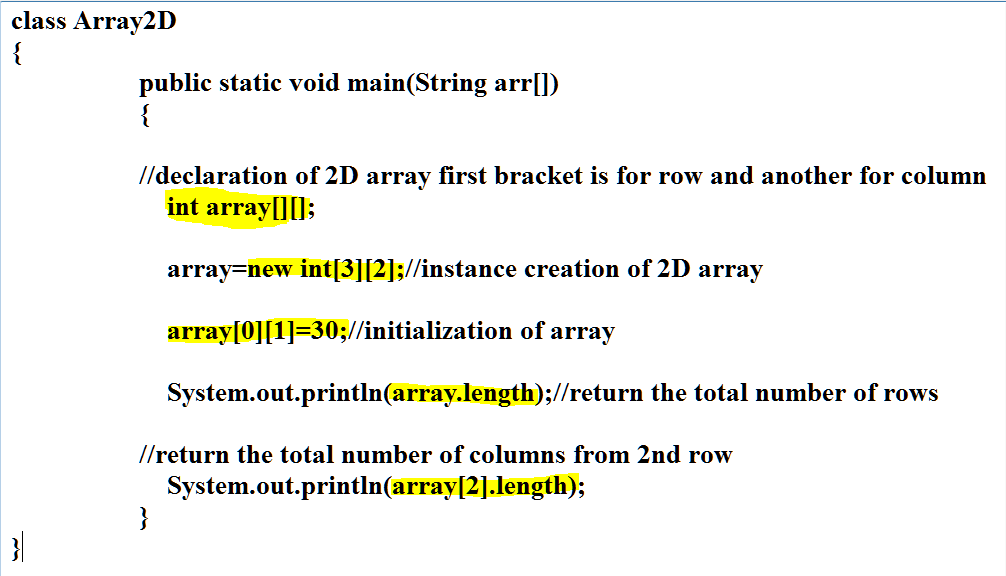
**Array**

1. Array is a collection of same type of values.
2. Array is fixed in size.
3. Array is indexed based
   1. Index is always start with 0
   2. Index is always -1 than size
4. If Do not intialize array then it will get initialize by default values.
5. Syntax
   1. Declaration of array(do not provide size at declaration time)
   2. Instance/object of array(must provide size for array)
   3. Initialization of array

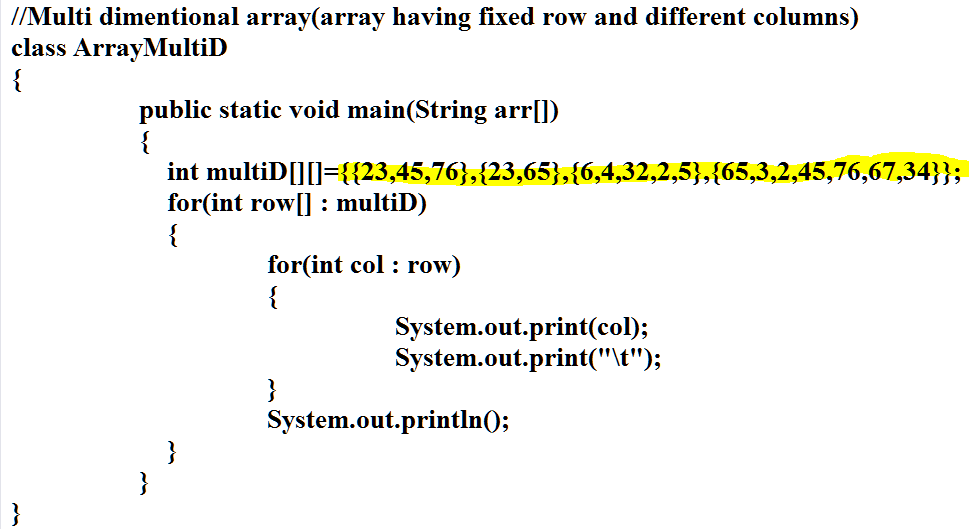




**Array 2D**

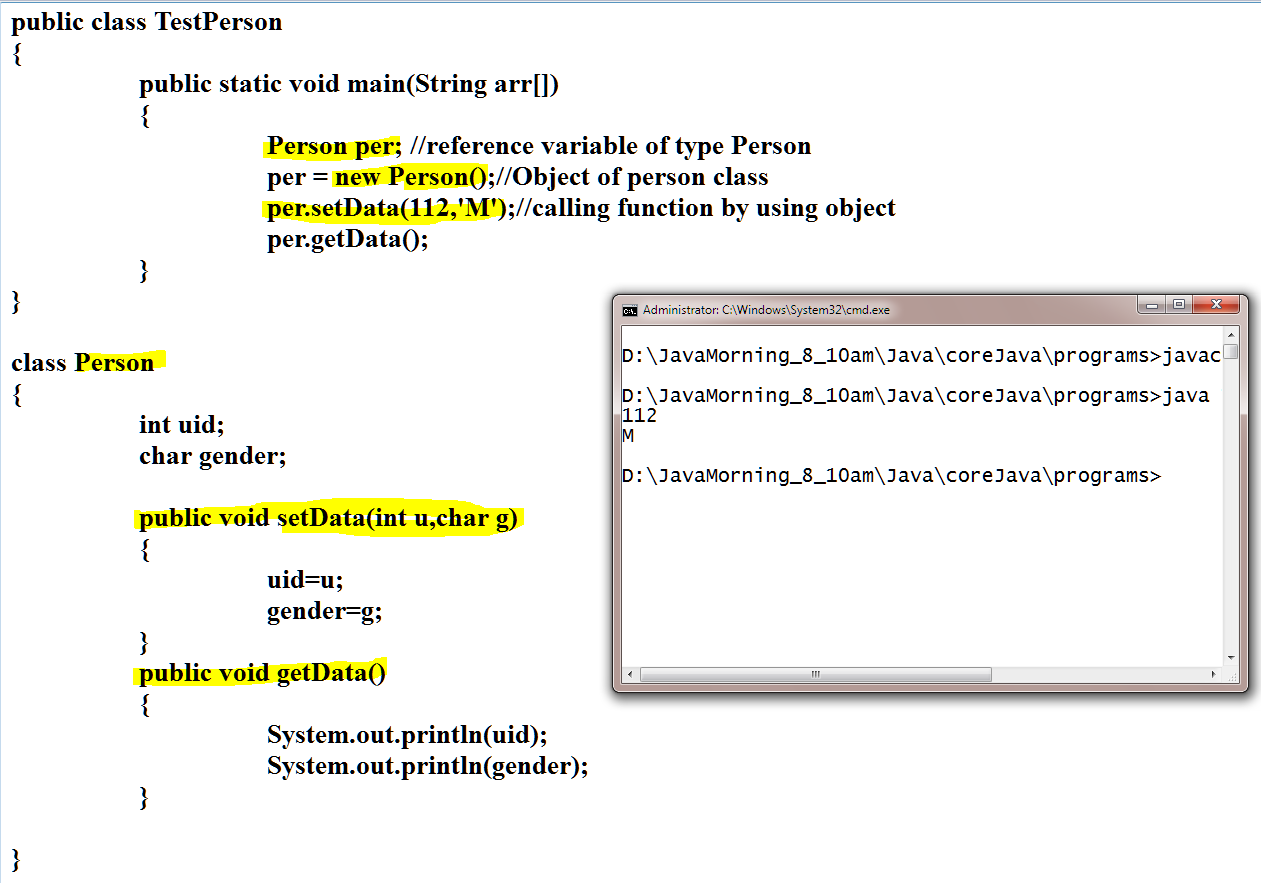


**Multi Dimentational Array**



Session 31-Jan-2020

Class and Object



Session 3 Feb 2020

Class :

1. Collection of state(variable/data memeber) and behaviour(methods/member function)

Object :

1. Object is a instance of class.
2. It is use to repersent class members.
3. Object can access members of class independantly.
4. Each object has its own existance.
5. Object gets memory inside heap.
6. Each object has its own copy of instance variable.
7. By using object can access any member of classs outside any other class if and only if, it is non

private.

1. Syntax for Object

To create Object, first we need the reference of Object, then can assign class object to the

reference variable.

**Syntax for creating Reference variavle :**

ClassName identifier(Name Of Object);

**Syntax for creating Object and assign object to reference variable :**

Identifier = new ClassName();

**OR**

**Above syntax can be write by following ways also :**

ClassName identifier = new ClassName();

**Namining rules and conventions for creating identifier**

**Rules**

1. Identifier must be start with any char or symbol \_ and $
2. Identifier can contains number but must not start with numbers.
3. Cannot use keywords(pre-define words bt java language) as a identifier.
4. Identifers are case sensitive.
5. There is no limit for number of char inside identifier.
6. Cannot use another things inside identifier except a-z, A-Z, 0-9, $ , \_

Example

MyClass : ValidZ

Int : Valid

Void : Valid

1stNumber : invalid

\_1234 : Valid

$\_ : Valid

$demo : Valid

Email@User : invalid

User\_Name : Valid

PublicData : Valid

public : invalid

**Conventions to create class identifier**

1. Class name should be start with upper case.(First char shuold be upper case)
2. If class name is a combination of more than one word then each word first

Char should be start with upper case.

1. Class name must follow all the identifier rules.
2. Example
   1. Employee
   2. ProductDetails
   3. EmployeePersonalInfo

**Conventions to create variable and method identifier**

1. Variable and method name should be start with lower case.
2. If variable or method name contains more than one word then 1nd onwords

Word first char should be upper case.

1. Cariable and method name must follow all the rule
2. Example
   1. setData
   2. employeeId
   3. printEmployeeDetails

**Conventions to create constants identifier**

1. Contant name must be in upper case.
2. Must follow all rules.
3. Example
   1. PI
   2. COMPANY\_NAME

Session 4-Feb-2020

**Build-In class :-** class which is pre define inside the language

**Custom class :-** the class created manually.

**String class**

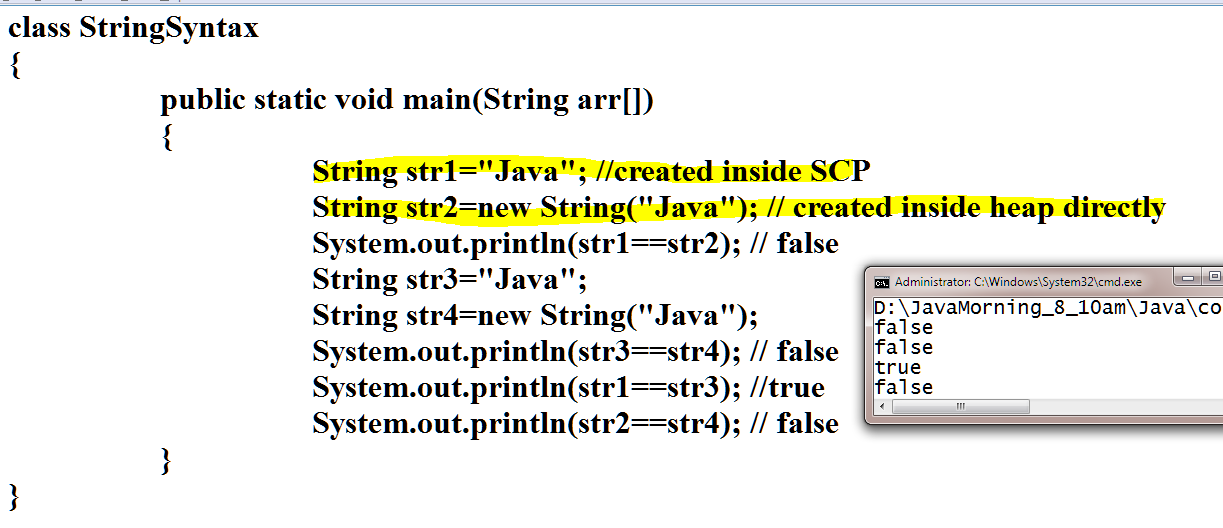
1. String is build in class.
2. String can store array of char.
3. String is immutable object.(original value of String reamains same after applying any function of String)
4. String is in java.lang package.
5. String is a final class.
6. String has functions using which can perform operations on existing string.
7. Syntax to create Object of string.

String str = ”Value”;

String str= new String(“Value”);

1. The String created without new operator goest into String Constant Pool(SCP).
2. While storing values iside SCP first check values are already exist or not if

Value id alredy exist then not get new memory else get new memory.



Session 5-feb-2020

**SCP(String contant pool)**

1. SCP is a part of heap where string objects gets created.
2. The string object created without new operator gets memory inside SCP.
3. Before allocating memory for any string value inside SCP, it will check for already same value exist or not, if same value is exist then will not create new memory else create new memory.
4. Feb-2020

OOP’s Concepts

**Encapsulation :**

What :

1. Wrapping of data member(variable) and member functions(methods) into single unit.
2. Class is also an example of encapsulation.

How to implement :

1. By Declaring instance variable as a private
2. And provides access to variables by setters and getters method.

Advantages :

1. Can achive data hiding.
2. Can achive Loose coupling(If change one part of code it will not affect another part).
3. Easy to add new requirements.
4. Can be use for data validations also.
5. Reduce repetation of code(like data validation code).

Example/Syntax

