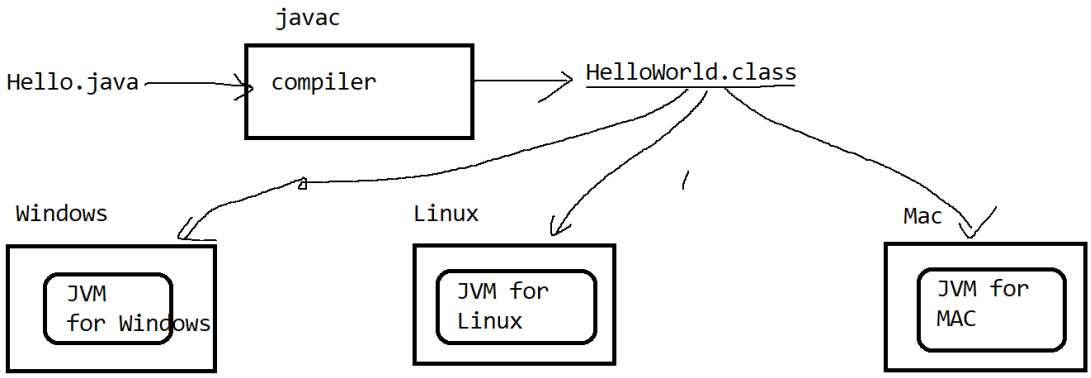
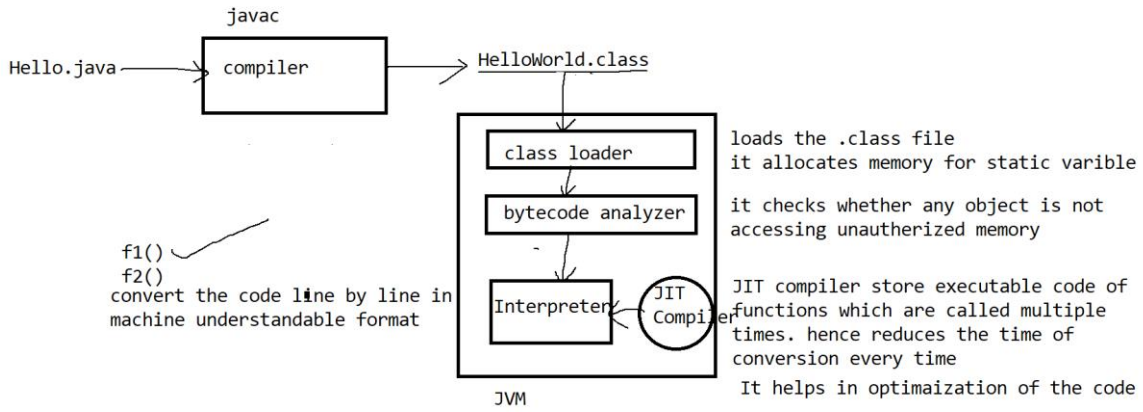
Object oriented Programming using java

Features of java

- Simple : because it doesnot have * operator
- Platform independent: the code written in windows machine can run on linux or Mac machine also.
- Machine independent-→ it is underlying h/w independent
- High level language→ in java we cannot access low level memory by address in the program.
- Multithreading is possible
- Networking is possible
- Data base connectivity is possible
- GUI programming is possible
- Garbage collector facility is available-→ so no memory leakage problem.
- Compiler and Interpreter
- Object oriented.



JVM



System, Integer, String all these are classes in java.lang library, hence no import is required

To store all .class files in separate folder by name bin

C:\javademos> javac -d bin HelloWorld.java

To run the file

C:\javademos> java -classpath .\bin HelloWorld

To avoid adding classpath for every execution, we may set classpath

C:\javademos>set classpath=.\bin; C:\Program Files\Java\jdk1.8.0_162\lib

Basic data types

byte > short > int > long >float > double

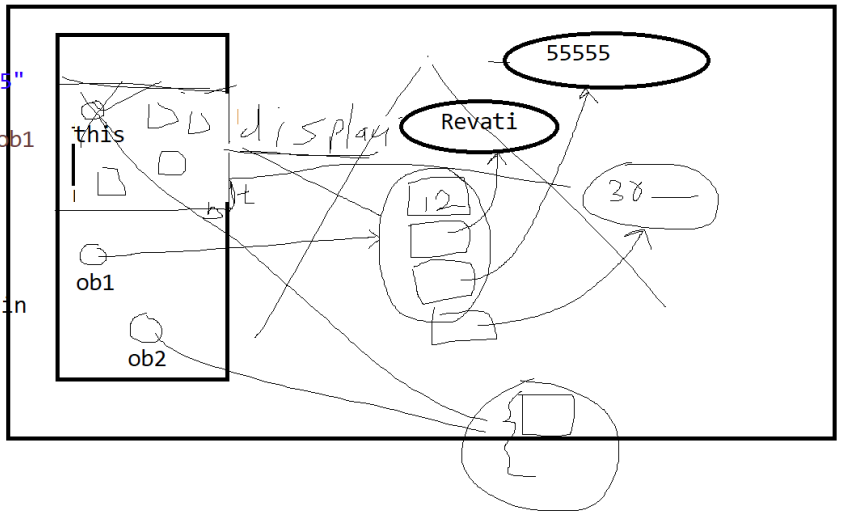
basic operators --→ +, -, /, %, *, ++, --

Relational operators -→ > , <, >=, <=, == , !=

Logical operators-→ &&, ||, !

Bitwise operators→ >>, <<, &, |, ~

```
//parameterized
constructor
    Person ob1=new
Person(12,"Revati","5555"
,new Date());
    System.out.println(ob1
);
```



Day3

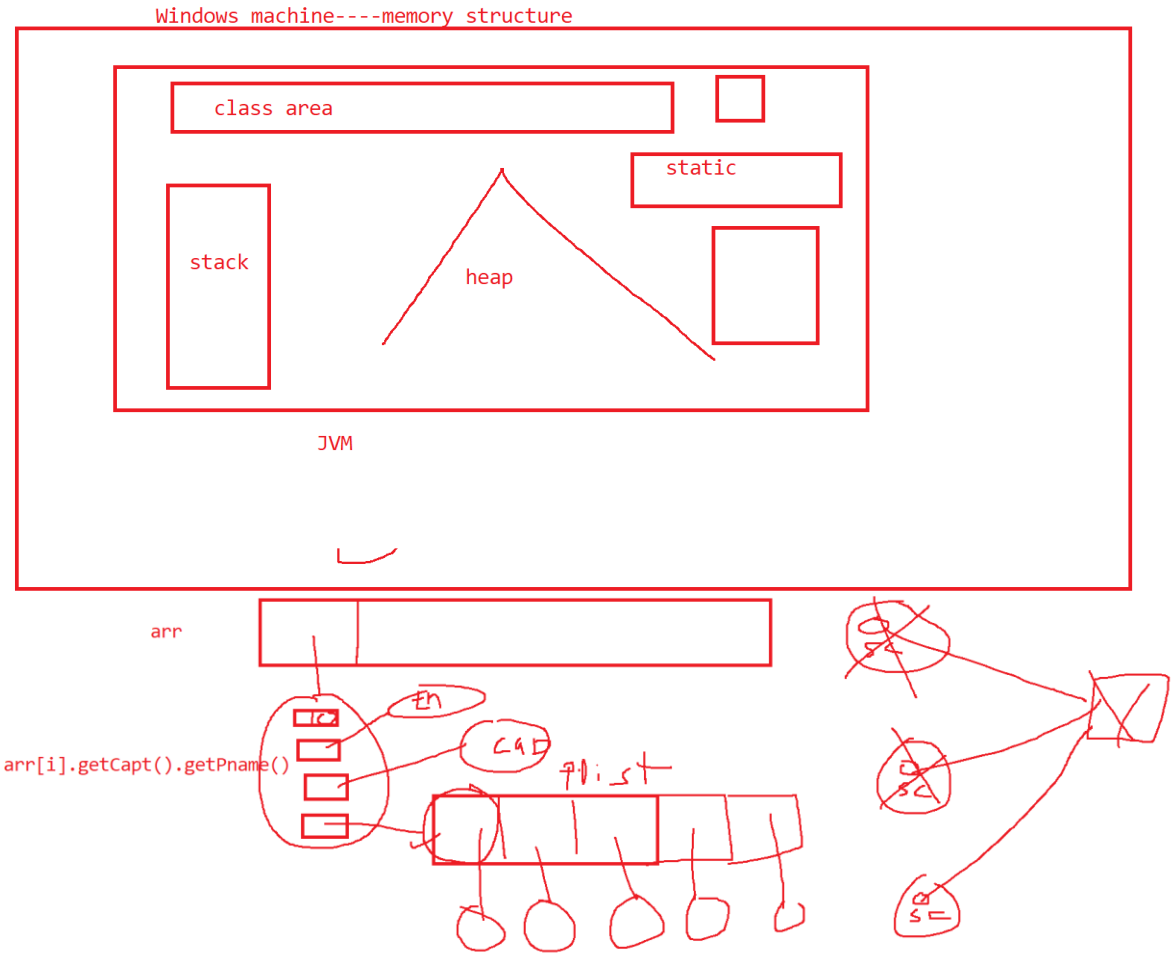
1. String class object is immutable,
2. Internally JVM maintains a pool of constants.
3. If you assign any constant String to a object then first it checks whether it exists in the pool. If it is there, then new object will not get created and all references will point to same object
4. If we use new String("test") constructor to create a object, always new object will get created
5. To create mutable String objects, use StringBuffer and StringBuilder, StringBuilder is used in single Thread application, because it is not thread safe
6. StringBuffer is thread safe so used in Multithreaded application.
7. Methods in String Buffer are not efficient as compared to StringBuilder class methods.

Classes are related to each other by either ISA relationship or HasA relationship.

HasA relationship is also called as aggregation , Aggregation are of 2 types

1. Composition→It is tight coupling between classes. Hence usually represented as nested classes.
   The class written inside the other class is called as nested classes
   The inner class can be a normal class, or it can be a static class.
   Outer class can never be static.
2. Association--→It is loose binding. So, both objects should have separate existence

Team class structure



Day 3

Every primitive data type has corresponding class called as Wrapper class, These wrapper classes helps to conver primitive data type into objects

In java 1.5 onward boxing and unboxing is implicit.

Convert object to primitive variable --- unboxing

Integer ob=5

int x=ob

Convert primitive variable to Object ----boxing

int i=4;

Integer ob=I;

byte	Byte
short	Short
Int	Integer
long	Long

float	Float
double	Double
char	Character
boolean	Boolean

To declare a variable as constant we use final keyword

Final variable has to be initialized at the time of declaration, but if it is a member of a class, the it can be declared and later initialized in constructor is allowed but only once;

Function overloading

Within a class if you write multiple functions with same name, but different number of parameters or different type of parameters then it is called as function overloading. It is also called as static polymorphism, because which function to call is known at compile time.

Static variables

A function can be static

- If you want to call the function without creating object then make the function static
- It does not receive this a s a parameter, hence cannot access instance variables
- It can access only static members of the class.

A variable can be static.

- If you want to share the memory among all objects of the class
- If you want to allocate memory and initialize it before creation of first object.
- These variables can be initialized in static block.
- Static variables memory will get allocated and inititalized as soon as class gets loaded in the memory.