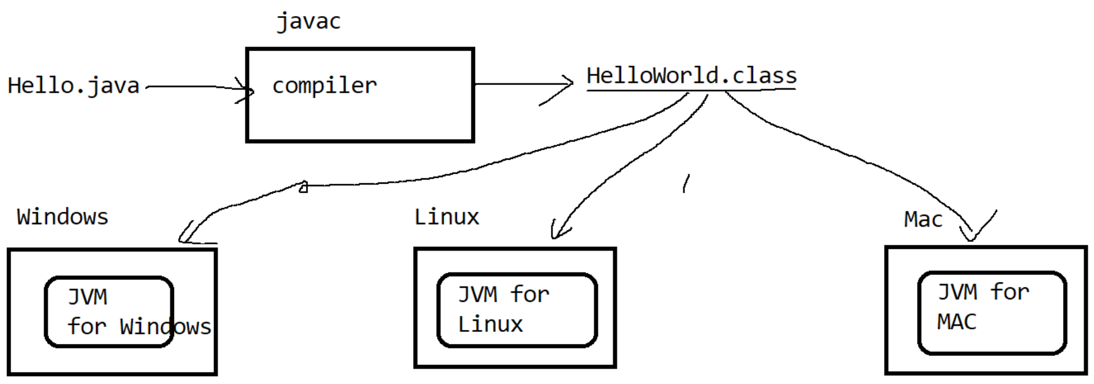
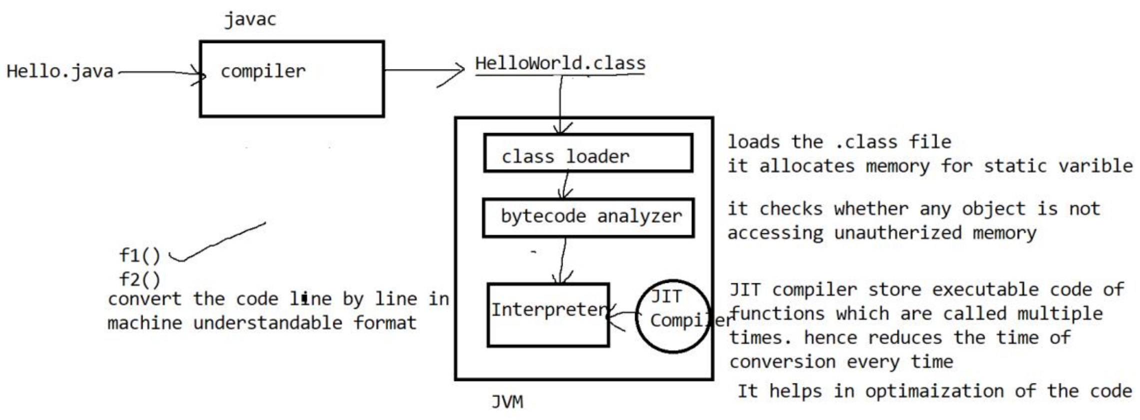
Object oriented Programming using java

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


javac
Hello.java
compiler
HelloWorld.class
class loader
bytecode analyzer
Interpreter
JIT Compiler
JVM
loads the .class file
it allocates memory for static varible
it checks whether any object is not accessing unautherized memory
f1()
f2()
convert the code line by line in machine understandable format
JIT compiler store executable code of functions which are called multiple times. hence reduces the time of conversion every time
It helps in optimaization of the code

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

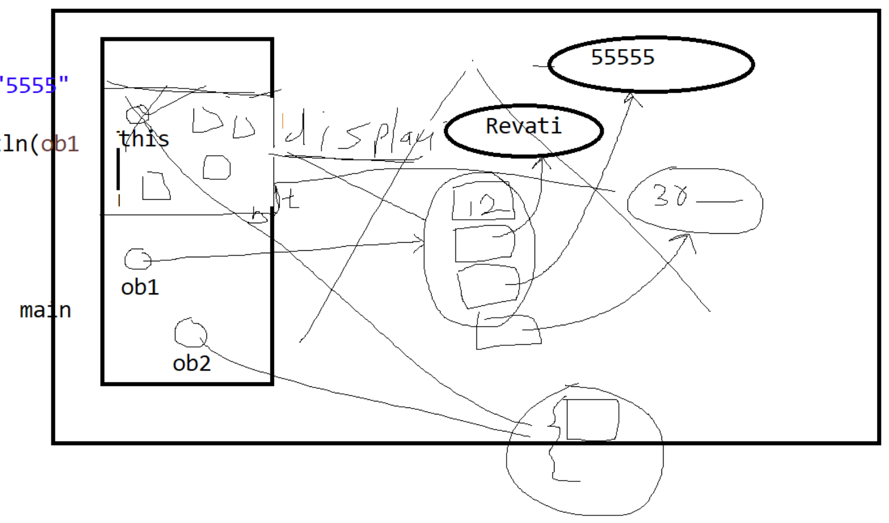
basic operators --→ +, -, /, %, *, ++, --

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

Logical operators-→ &&, ||, !

Bitwise operators→ >>, <<, &, |, ~

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



Day3

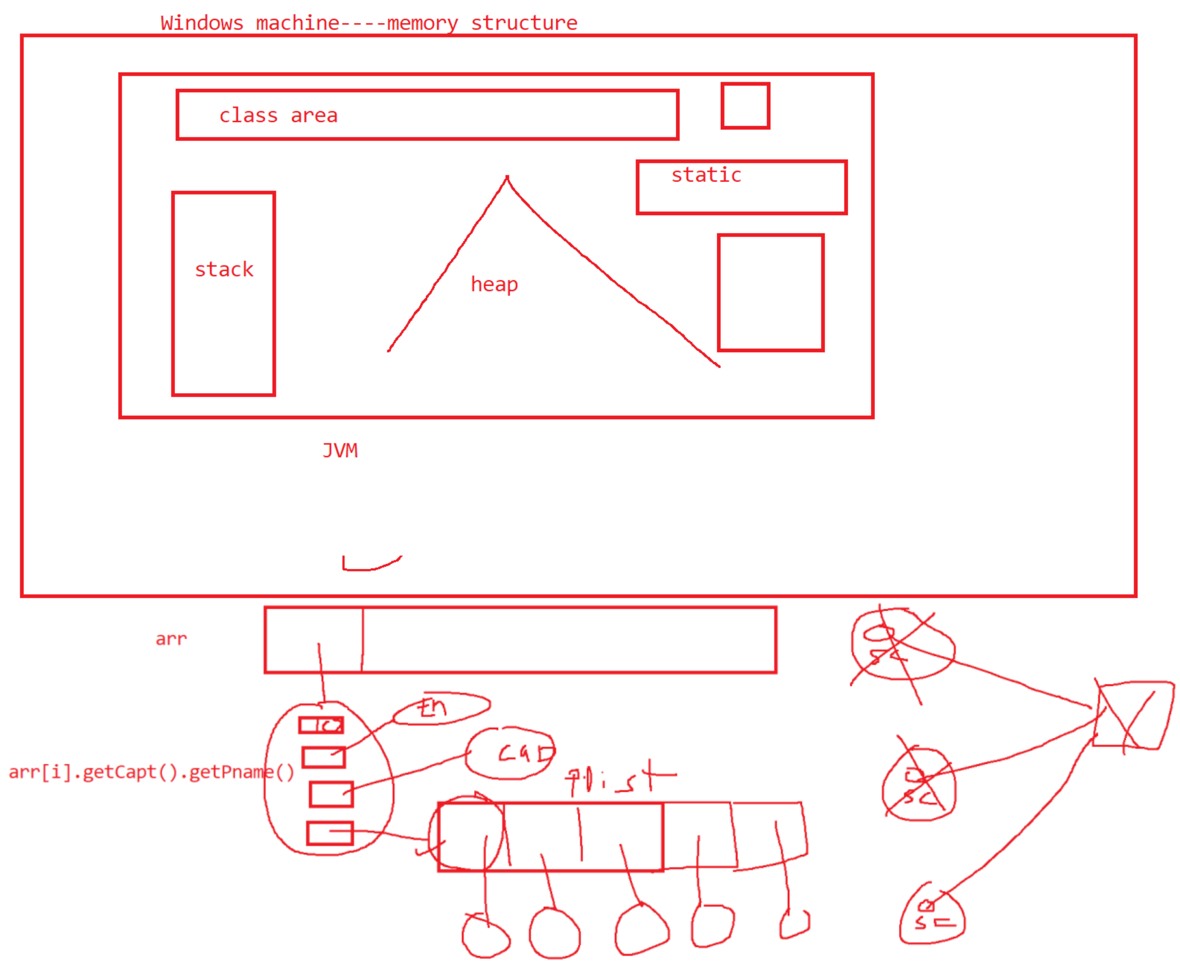
1. String class object is immutable,
2. Internally JVM maintains a pool of constants.
3. If you assign any constant String to a object then first it checks whether it exists in the pool. If it is there, then new object will not get created and all references will point to same object
4. If we use new 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


Windows machine----memory structure
class area
static
stack
heap
JVM
arr
arr[i].getCapt().getPname()
plist

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.

Packages in Java

Packages are folders in java project

Packages are used for

1. Better organization
2. Importing related classes becomes easy
3. To avoid naming conflict.

Access specifiers

Private- private members are accessible only within the class

Protected-→ protected members are accessible within all classes in the same package, and child classes.(within package/ outside package)

Public-→ public members are accessible within class, outside class everywhere.

In inheritance Hirarchy upcasting is implicit , and down casting in explicit.

Inheritance

It represents ISA hierarchy.

When you want to design multiple classes, and few members are common, then you can add them in parent class and inherit them into child class is called as inheritance

Inheritance is of 4 types

1. Single 2. Multilevel 3. Multiple 4. Hybrid

Java supports Single and Multilevel inheritance.

a. What is function overloading

If a class has more than one method with same name, but it has different number of parameters or different types of parameters, then it is called as function overloading.

It helps client, to remember less number of functions

If a parent class and child class has same method woth same name and same number of parameters and same return type. Then it is called as function overriding.

Function overriding helps in dynamic polymorphism, it helps to reduce the size of code.

Parent class reference can point to child class object → it is upcasting

Upcasting is implcit

Person p=new Student()

Child class reference pointing to Parent class object -→ it is downcasting

Down casting is explicit

Student p=(Student)new Person()

Interfaces VS abstract classes

Interfaces	Abstract classes
Interface is contract between classes and interfaces	It is used to represent ISA relationship
All the methods by default public and abstract. Static methods can be added. (1.8 onward) If the method has implementation then it should be default method, these methods can be overridden, but it is optional(1.8 onward)	It contains minimum one abstract method, but also may contain method s with implementation
Constructor is not there	Constructor is there
All variables are by default public static and final	Variables may be static or instance variables
One interface can extend any number of interfaces	One class can extend only one class and can implements any number of interfaces

Collection Classes

List

It is an ordered collection, means it stores the data in the same sequence in which we entered.

Since it is ordered data can be retrieved randomly, by using index.

Duplicate values are allowed to store.

Vector:

It is legacy class.

It internally uses arrays, it is ordered

All the methods are synchronized hence good to use in multithreading.

ArrayList:

It internally uses arrays. It is ordered collection.

LinkedList:

It internally uses linked list.

ArrayList functions which needs to search the object based on few members of the class internally call equals function.

example

indexOf, contains, remove, lastIndexOf, retain, retainAll, removeIf

hence override.

boolean equals(Object ob) in employee class