JAVA

Public

→ A Java file can have many classes

→

OOPS Concepts

1) Encapsulation:

→ data members of a class should not be directly accessed outside of the class.

→ They can be used by using member functions of the class

Static key word - A variable having "static" keyword will have memory allocated by JVM before the execution of the class.

Primitive data types

void	(Void)	-- null
boolean	(Boolean)	-- 1 bit
byte	(Byte)	-- 8 bits
char	(Character)	-- 2 bytes
short	(Short)	-- 2 bytes
int	(Integer)	-- 4 bytes
long	(Long)	-- 8 bytes
float	(Float)	-- 4 bytes
double	(Double)	-- 8 bytes

method area
↓
static

object Creation:

"new" → requests JVM for the creation of new dynamic memory area (object)

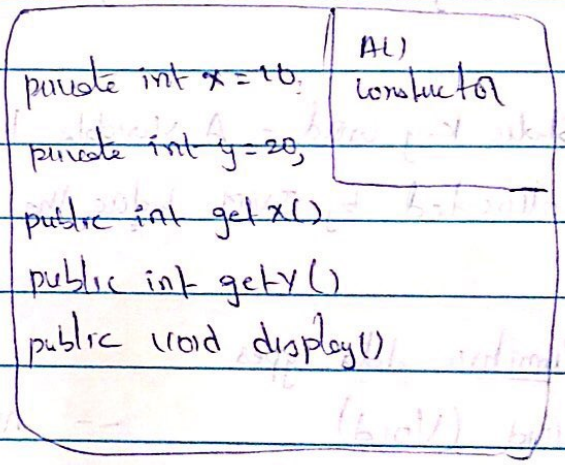
Constructor name after the new operator → to which class the memory to b allocated.

JVM PARTITIONED RAM

STACK

Garbage collected HEAP

new A()



2/25/18

In every software there will be two types of files

1) Binary → files which are used to compile & execute & run developed app
2) Library Files → files which are used to develop appl. (supporting files)

environmental Variable:

Predefined variables of OS which store the URL of binary & library files and will notify the OS about the location of the binaries and libraries of any installed softwares

1) Path – Binary files location
2) class → Library files location

compilation → compiler [C: program Files: Java: Jdk: bin: Javac.exe]

D: Javac.exe A.Java

A class file is generated with A.class which will have byte code of A.Java
[If the code is sylatically correct]

Interpreter :- will expect .class file with pre signatured main method.

3/31/18

* class scope variables are given default values if not initialized

```
private int x;  // default 0
```

* Local variables are not given default values

```
public void display(){
    int z;
    print(z) // error
```

(method)

we cannot use private variables in public methods

4/1/18 Constructors

* Never make a static before a normal java class

4/29/18

Static

static block → If we want to do somethings before execution of the class (pre-activities) then we will use static blocks.

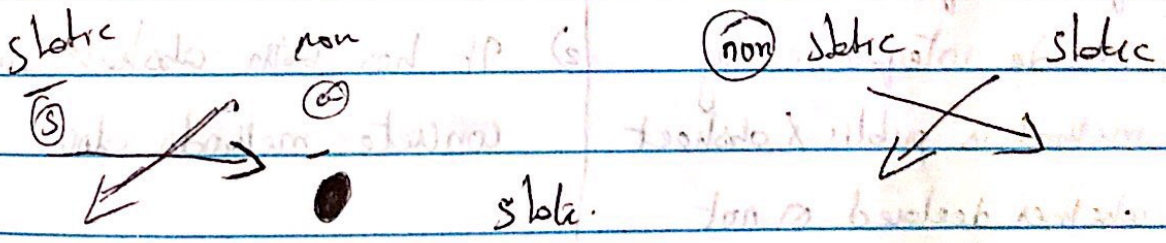
→ It will be discussed before main method.

* non-static members cannot be called in static context directly.

Static block	non-static block	At the time of object creation then the non-static blocks will be executed.
static {	{	
}	}	

static variables are stored in method area.

this & super are used to call non-static objects.



Overriding

Inheritance

we should not write

find Variables can be overridded but

find methods cannot be overridded

things

Wrapper classes

Interface	Abstract class
1) If we don't know anything about implementation, just we have requirement specification, we should go for Interface	1) If we are talking about implementation then we should go for Abstract class
2) Inside interface, every method is public & abstract whether declared or not	2) It has both abstract and other concrete methods also.
3) every variable present inside interface is always public, static and final whether declared or not.	3) Not required.
4) for variables in interface we have to initilize the values at the time of declaration.	4) Not required
5) Can inherit multiple interfaces	5) Can inherit only one parent class

5) we cannot declare instance and static blocks in the interface 6) we can't declare constructors	5) We can 6) we can

→ Abstract class is for local scope.
→ Interface is used for global scope. } Imp

Imp *

```
Snake snake = new snake();
snake.move(); ✓
snake.sleep(); ✓
```

```
Animal an = new Snake();
an.move(); ✓
an.sleep(); X
```

Imp

we cannot call local methods of sub class when class is created with parent class reference.

```
Animal {
    move();
}

Snake extends Animal {
    move();
    sleep();
}
```

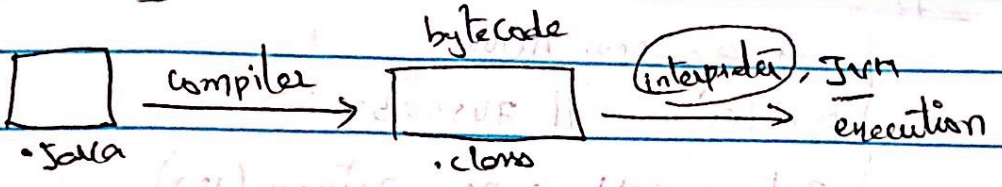
Java 8 (multiple interface doubt)
& default methods with same name.

Q

→ static methods of either ~~involved in~~ the interface are not involved in inheritance.

→

Exception Handling



checked	unchecked exceptions
showing exception before compiling. File not found exception	Run. time

class A {
int x;
}
A(x) {
this.x = x
}

Wrapper class

```
A a = new A(100)
print(a) = // 745ea62 ->
Integer int1 = new Integer(100)
Print int1 = 100
```

Boxing:

Storing primitive data type value in its corresponding wrapper object is known as Boxing.

```
Float floatObj = new Float(3.5f); // Boxing
Float objl = 3.5f; // Auto Boxing.
```