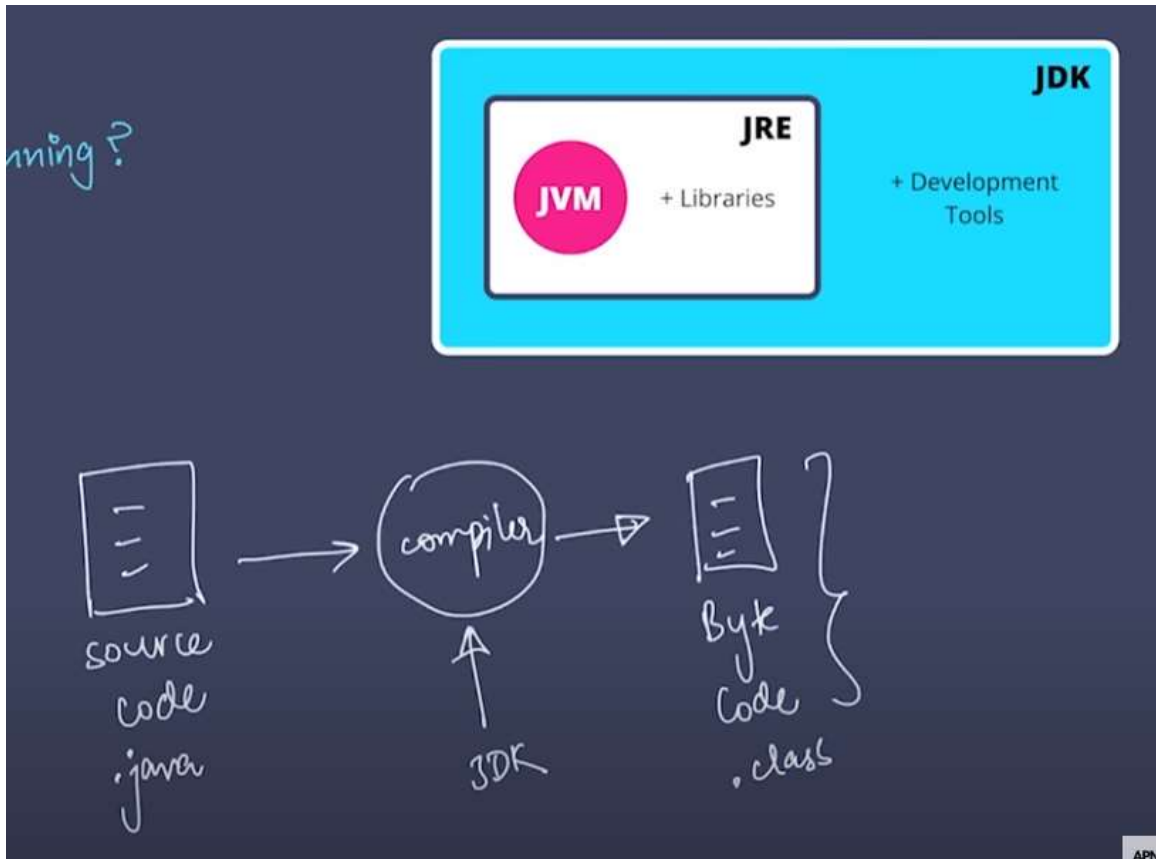


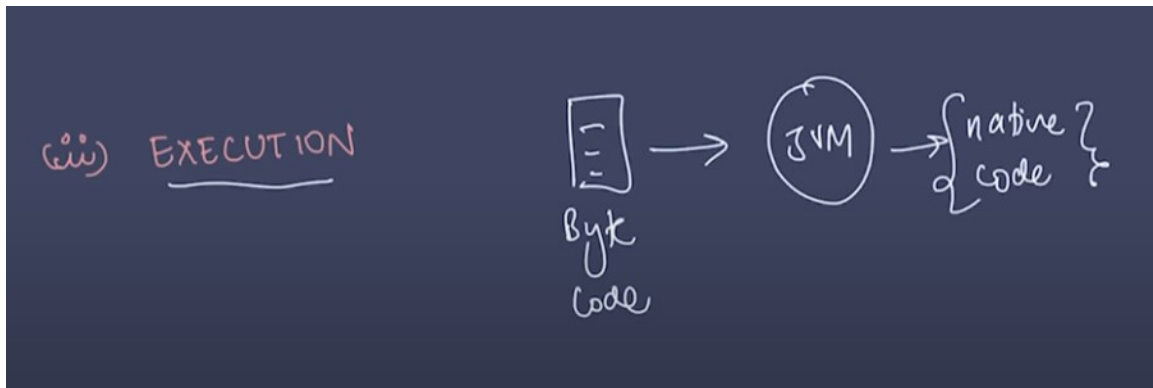
## 1) Compilation:

source code converted to byte code with the help of compiler(jdk)



## 2) Execution

Byte code is then converted to native code(understandable by machine)



Functions and Methods :

## Java Class 1

### 1. Function

```
void main() {  
  
}
```

### 2. Class

```
class Main {  
    void main() {  
  
    }  
}
```

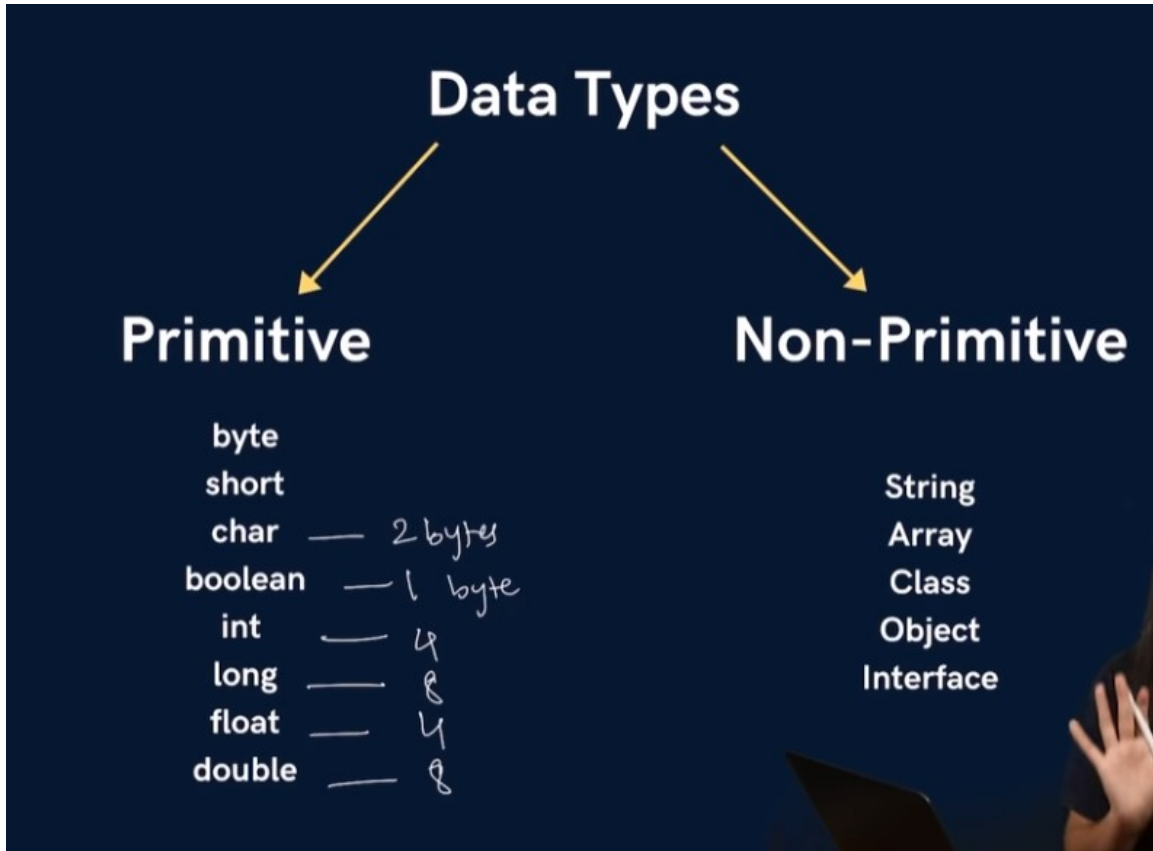
**void** : empty (does not return anything) , here main is the function.

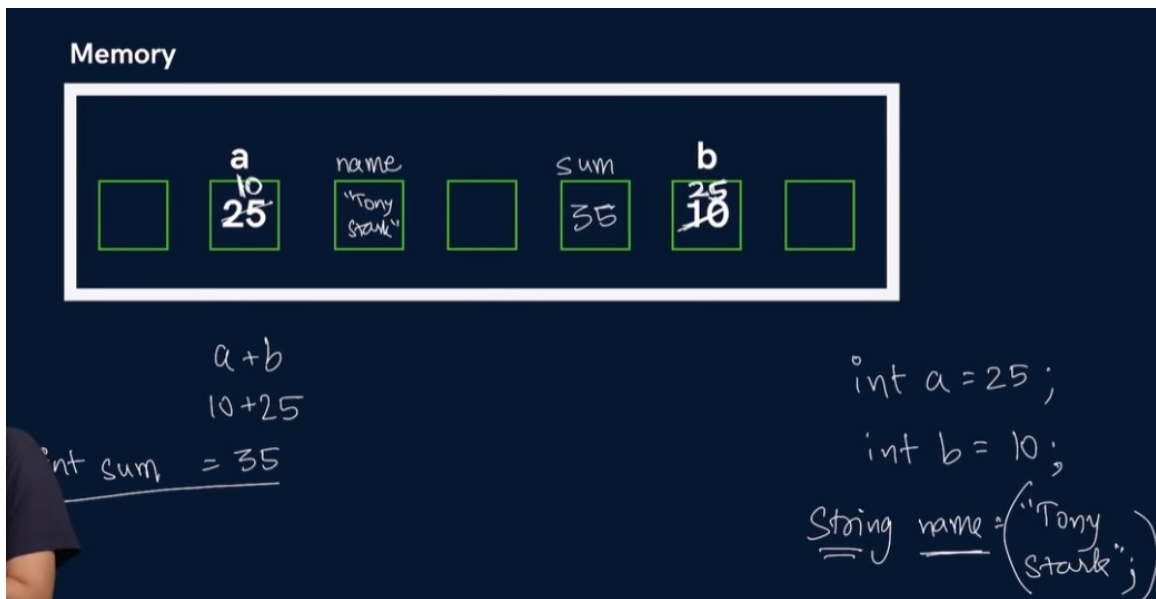
**class** is the entity which contains multiple functions. In java code , in class main function is the first function which gets executed .

\* Difference between Print and Println : ( Print ) prints the output in same line

Println : Statement inside println is printed and then program moves to next line .

\n : print in next line





\* Taking Inputs in java : we take help of Scanner class :

Conditional Statements :

# Conditional Statements

if, else

else if

switch



break

Switch Case :

```
switch (variable) {  
  case 1 :    stt 1;  
              break;  
  
  case 2 :    stt 2;  
              break;  
  
  case 3 :    stt 3;  
              break;  
  
  default :   stt 4;  
}
```

If case 1 becomes true , it then breaks out after case 1 , else proceeds to case 2 for execution and so on ....

**Loops :**

# Loops

for Loop

while Loop

do while Loop

```
for(initialisation; condition; updation) {  
    //do something  
}
```



```
while(condition) {
```

```
//do something
```

```
}
```

do {

```
//do something
```

} while(condition);

perform the statements till the while condition is satisfied.

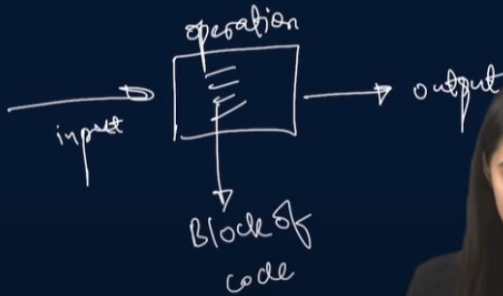
Whether what may be the condition , statement is executed minimum once.

**Functions and Methods :**




Block of code , Which -performs some operation by taking input and gives some output .

```
returnType functionName(type arg1, type arg2 .. ) {  
    //operations  
}
```



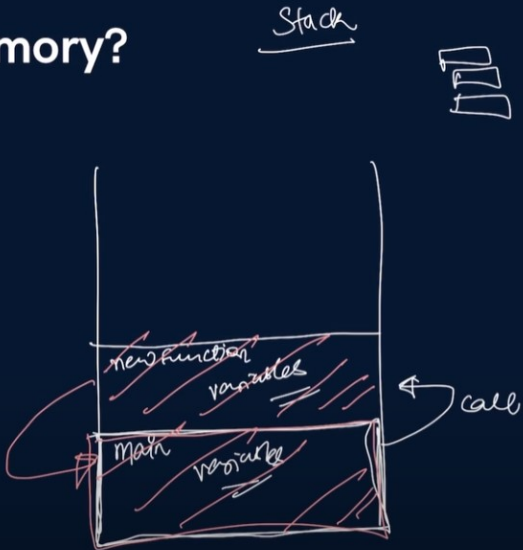
The diagram shows a box labeled 'operation' with an arrow labeled 'input' entering from the left and an arrow labeled 'output' exiting to the right. Below the box, an arrow points down to the text 'Block of code'.




A woman with long dark hair is sitting in front of a laptop, looking at the screen and gesturing with her hands.

## What happens in memory?

Stack



The diagram shows a vertical stack of memory frames. The bottom frame is labeled 'main' and contains 'main variables'. Above it is a frame labeled 'newFunction' and contains 'newFunction variables'. An arrow labeled 'call' points from the 'main' frame to the 'newFunction' frame. Another arrow labeled 'call' points from the 'newFunction' frame back to the 'main' frame. To the left of the stack, there is a list of code snippets: 'public static', 'main()', 'call', and 'newFunction'.



A woman with long dark hair is sitting in front of a laptop, looking at the screen and gesturing with her hands.

In memory : in first stackframe , main function variable are stored , when some function is called , one more stackframe is created for new function, when this new function returns value to main function , it gets removed and after execution of main function , first stackframe also goes off .

