Today's Agenda :-

→ Stack vs heap memory
→ Classes
→ Objects
→ Constructor
→ this keyword

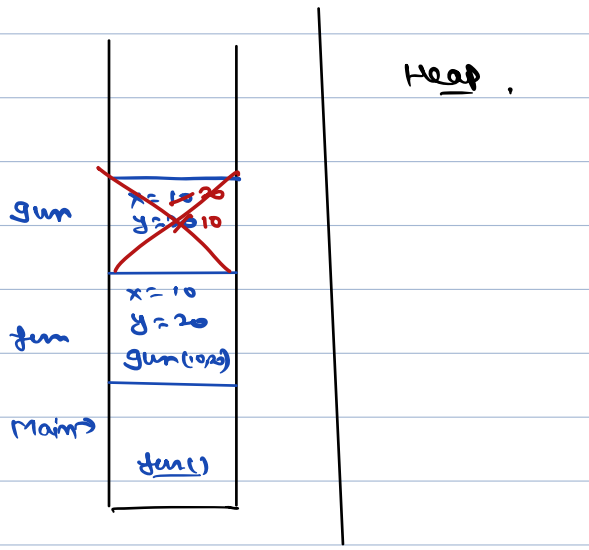
Today's Quote :-

what's the most important step a person can take ?.

—> The next step.

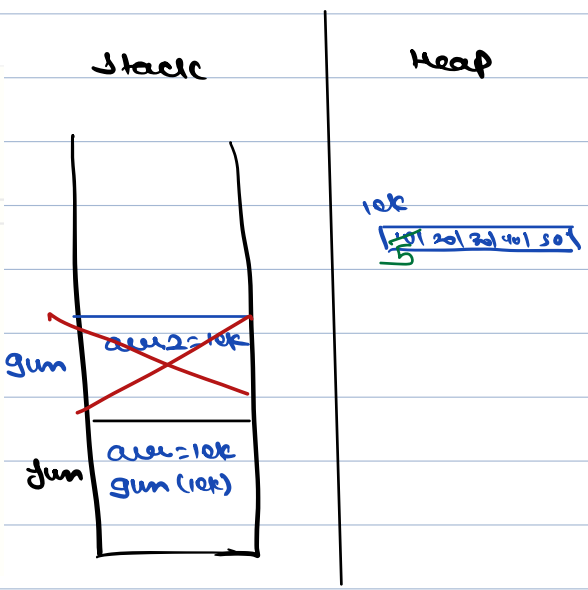
```java
class Main {
    public static void main(String args[]) {
        fun();
    }

    // Quiz 1)

    public static void fun(){
        int x = 10;
        int y = 20;
        gun(x,y);
        System.out.println(x+" "+y);
    }

    public static void gun(int x, int y){
        int temp = x;
        x = y;
        y = temp;
        // System.out.println(x+" "+y);
    }
}
```

2 Memory areas :-



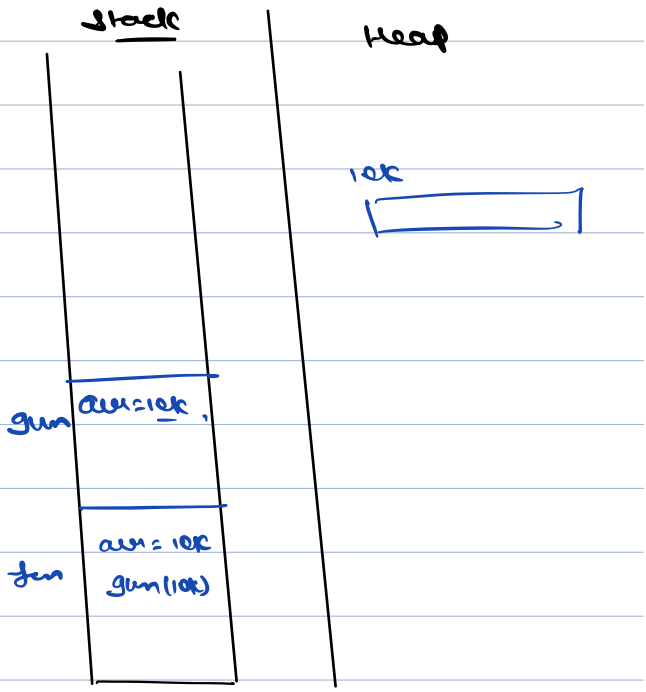

```java
class Main {
    public static void main(String args[]) {
        fun();
    }

    public static void fun(){
        int[] arr = {10,20,30,40,50};
        gun(arr);
        System.out.println(arr[0]);
    }

    public static void gun(int[] arr2){
        arr2[0] = 5;
        // System.out.println(arr2[0]);
    }
}
```

int[] arr = new int[5]
arr[0] = 10
arr[1] = 20
⋮

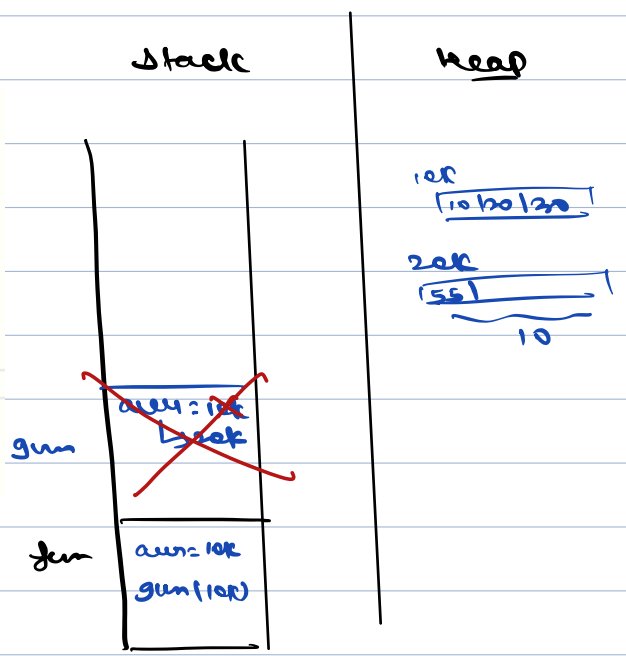


new → heap.

```java
class Main {
    public static void main(String args[]) {
        fun();
    }



    public static void fun(){
        int[] arr = {10,20,30,40,50};
        gun(arr);
        System.out.println(arr[0]);
    }

    public static void gun(int[] arr){
        arr[0] = 5;
        // System.out.println(arr2[0]);
    }
}
```




```java
    public static void fun(){
        int[] arr = {10,20,30,40,50};
        gun(arr);
        System.out.println(arr[0]);
    }

    public static void gun(int[] arr){
        arr = new int[10];
        arr[0] = 55;
    }
}
```



Class → Blueprint of an Object.

Object → It's an instance of a class.

Students :- Name
↓ Age
50 Psp

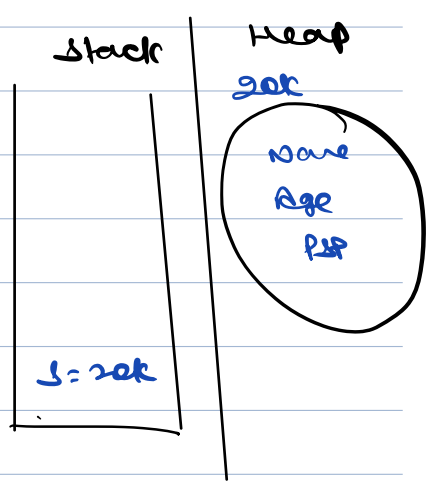
String [50] name; → name[0]
int[50] Age; → Age[0]
int[50] Psp; → Psp[0]

we can create our own data type, which we call class.

```
class Student {
    String name;
    int Age;
    int Psp;
}
```

Template.

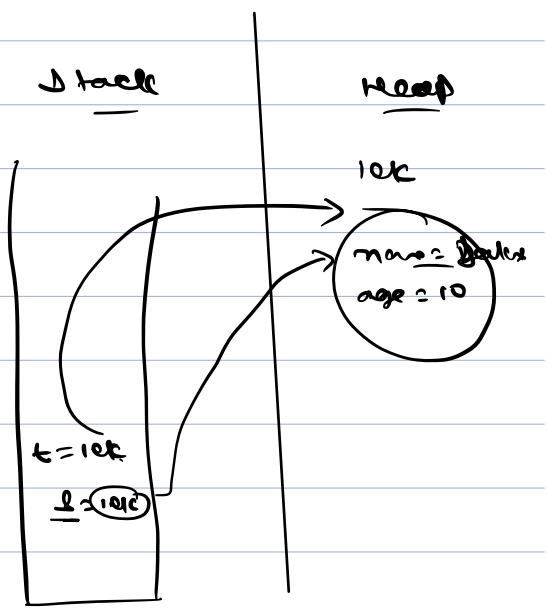
Student s = new Student();



```java
class Student{
    String name;
    int age;
}

class Main {
    public static void main(String args[]) {
        Student s = new Student();
        s.name = "Deepak";
        s.age = 10;
        System.out.println(s.name);
        System.out.println(s.age);
    }
}
```


```java
class Main {
    public static void main(String args[]) {
        Student s = new Student();
        s.name = "Deepak";
        s.age = 10;

        Student t = s;
        t.name="Saket";

        System.out.println(s.name);
        // System.out.println(s.age);
    }
}
```

int a



```java
class Student{
        String name;
        int age;

    // constructor -> function
    Student(){
        name = "TestUser";
        age = 100;
    }
}
```

Name must be same as class name.

It has no return type

Constructor

It is a function, used to initialize data members of a class.

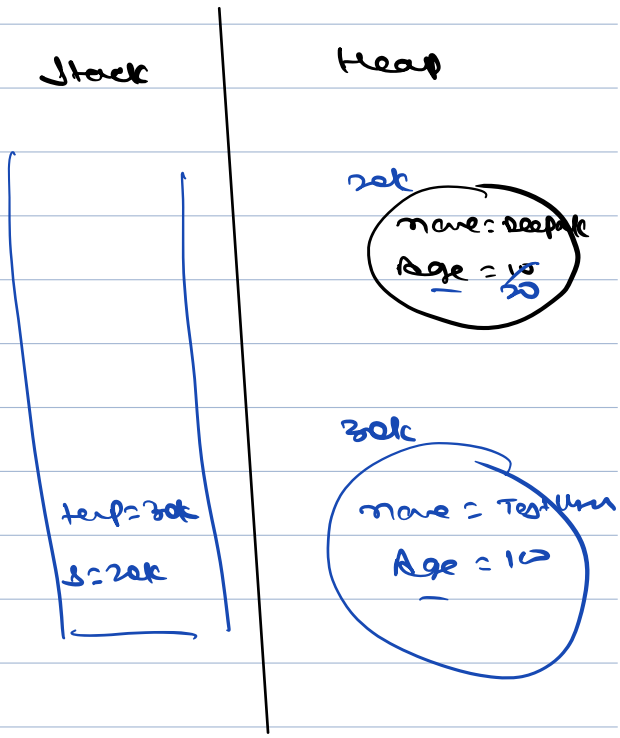
Student s = new Student()

You don't call it, as soon as object is created it is instantiated

called.

```java
class Student{
        String name;
        int age;

    // constructor -> function
    Student(){
        name = "TestUser";
        age = 100;
    }

    // Parameterised constructor
    Student(String a, int b){
        name = a;
        age = b;
    }
}


class Main {
    public static void main(String args[]) {
        Student s = new Student("Deepak",10);

        Student t = new Student();
        t.name="Saket";
        t.age = 40;

        System.out.println(s.name);
        System.out.println(s.age);
```

```java
class Student{
        String name;
        int age;

    // constructor -> function
    Student(){...
    }

    // Parameterised constructor
    Student(String a, int b){...
    }

    public void incrementAge(){
        age+=10;
    }
}
class Main {
    public static void main(String args[]) {
        Student s = new Student("Deepak",10);
        Student temp = new Student();
        // System.out.println(temp.name);
        // System.out.println(temp.age);
        s.incrementAge();
    }

    // s.name="Deepak"
    // s.age = 10

    // temp.name="testUser"
    // temp.age=100
}
```



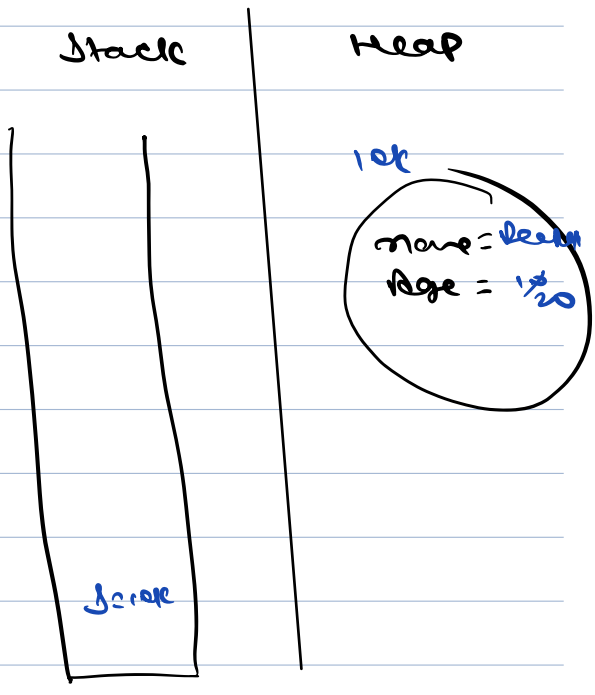
Break 10:16 pm – 10:26 pm

```java
class Student{
    String name;
    int age;

    // Parameterised constructor
    Student(String a, int b){
        this.name = a;
        this.age = b;
    }

    public void updateAge(){
        this.age+=10;
    }

}
class Main {
    public static void main(String args[]) {
        Student s = new Student("Deepak",10);
    }
}
```

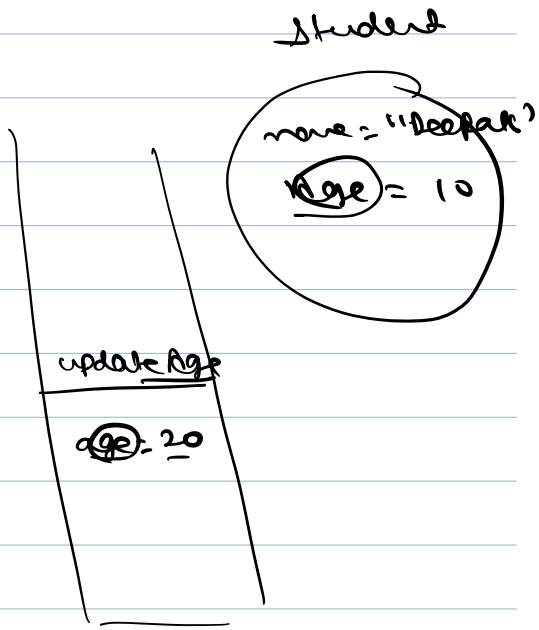
Deepak

10

s.updateAge();



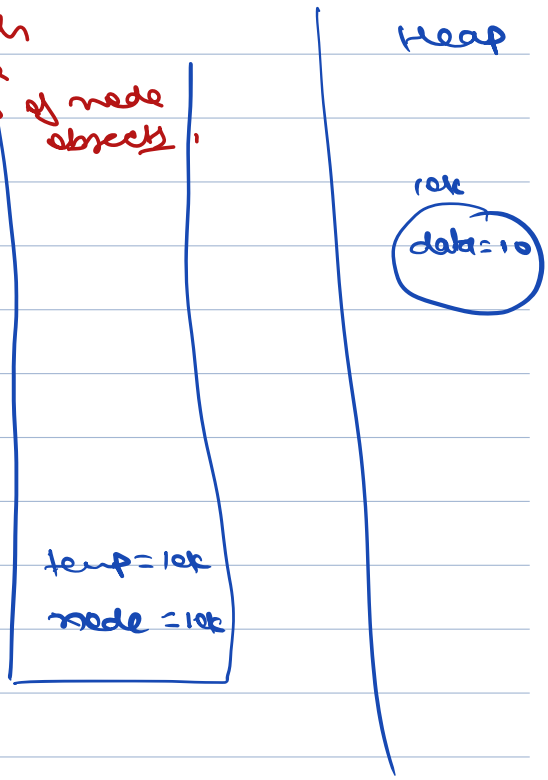
```java
class Student{
        String name;
        int age;


    // Parameterised constructor
    Student(String name, int age){
        this.name = name;
        this.age = age;
    }

    public void updateAge(int age){
        this.age+=age;
    }

}
class Main {
    public static void main(String args[]) {
        Student s = new Student("Deepak",10);
        s.updateAge(20);
        System.out.println(s.name+" "+s.age);
    }
}
```



→ datatype ;

```java
class Node{
    int data;
    Node next;
    Node(int data){
        this.data = data;
    }
}

class Main {
    public static void main(String args[]) {
        Node node = new Node(10);
    }
}
```

variable which can store address of node object.



Node temp = node;

int x;

String s;

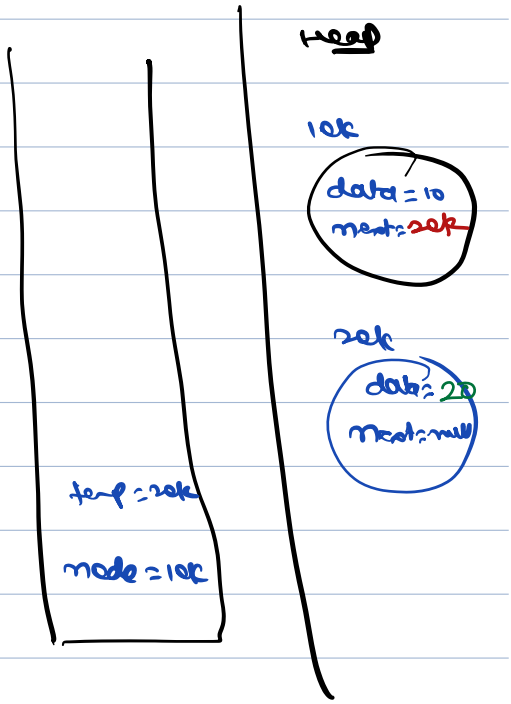
Node kalyan; → It can store address of node object.

Node Prabir;

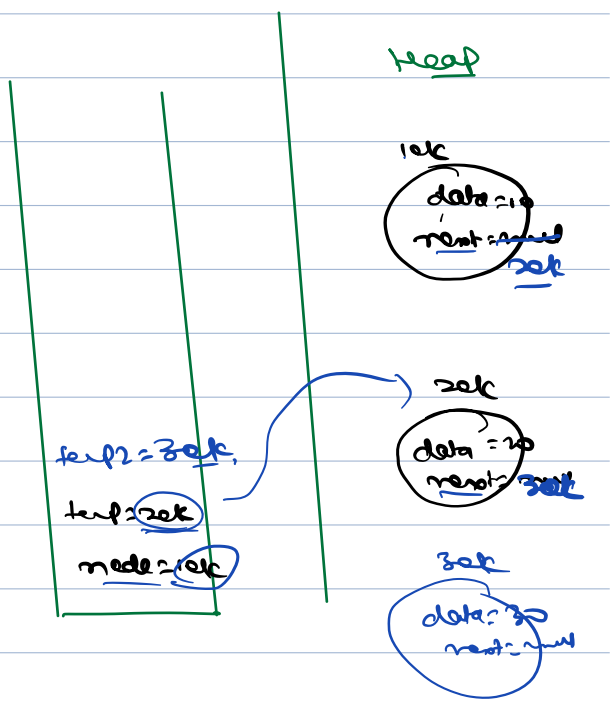
```java
class Node{
    int data;
    Node next; // next is a variable which can store address of
    // Node objects in heap

    Node(int data){
        this.data = data;
    }
}

class Main {
    public static void main(String args[]) {
        Node node = new Node(10);
        // node = 10k

        Node temp = new Node(20);
        // temp = 20k

        node.next = temp;
    }
}
```

20

print (node . next . data) ;




```java
class Node{
    int data;
    Node next; // next is a variable which can store address of
    // Node objects in heap

    Node(int data){
        this.data = data;
    }
}

class Main {
    public static void main(String args[]) {
        Node node = new Node(10);
        // node = 10k

        Node temp = new Node(20);
        // temp = 20k

        node.next = temp;
        // System.out.print(node.next.data);
        System.out.println(node.next);
        System.out.println(temp);

        Node temp2 = new Node(30);
        temp.next = temp2;

    }
}
```



node . next . next . data .

→ 30

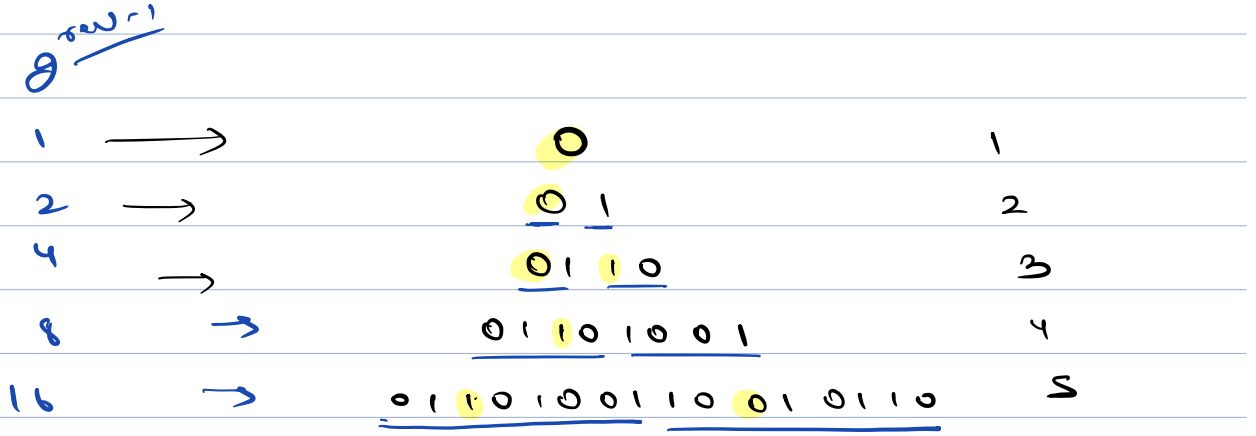
→ node . next . next . next . data ; → null pointer Exception.

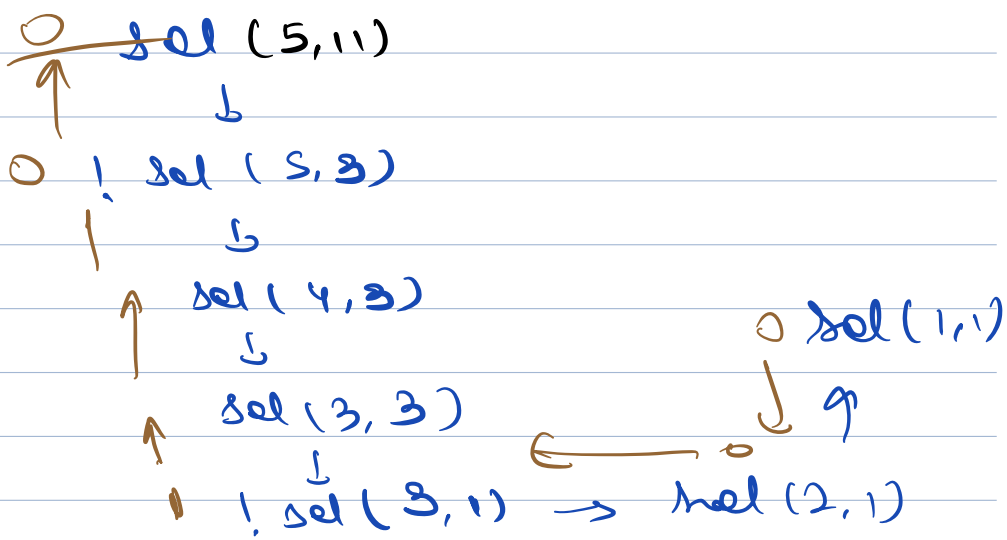
Student s;

int* arr = new int[10];

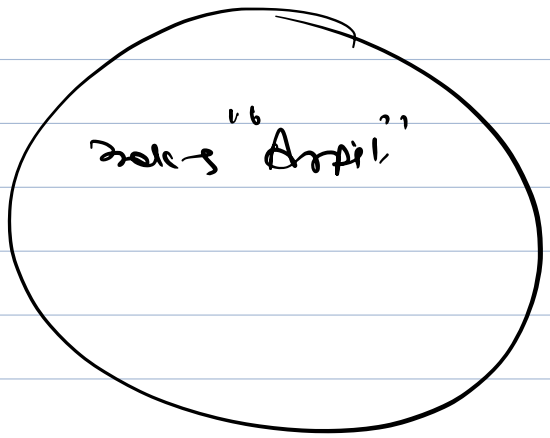
int arr[10]

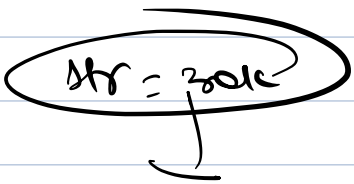
Static





String str = "Arpit"



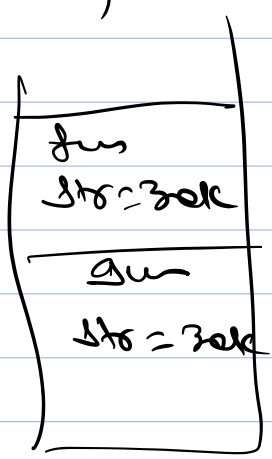


fun

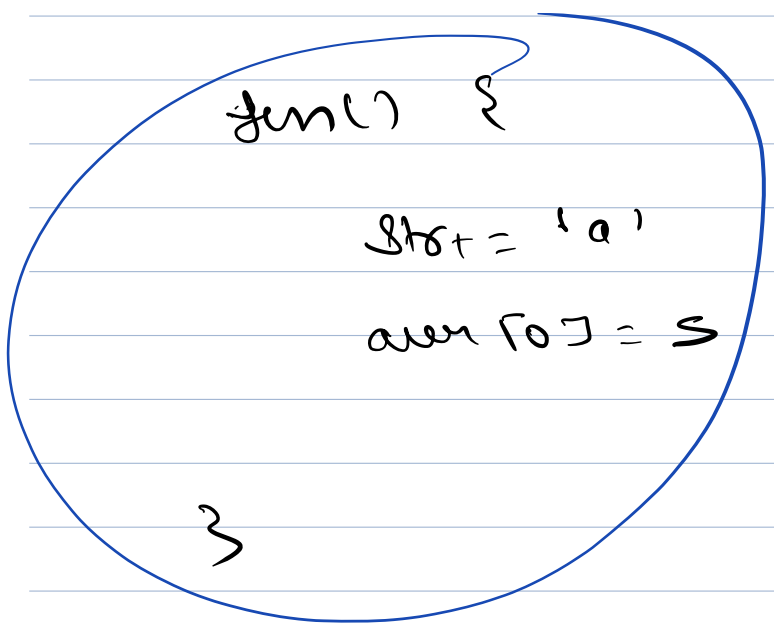
```
String str = 'Arpit'
fun ( str);
Print (str);
```

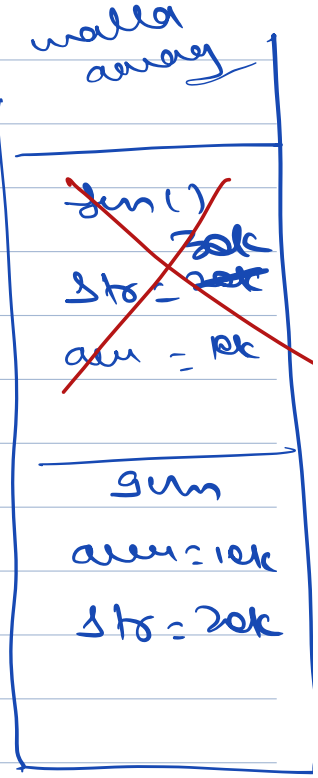
fun

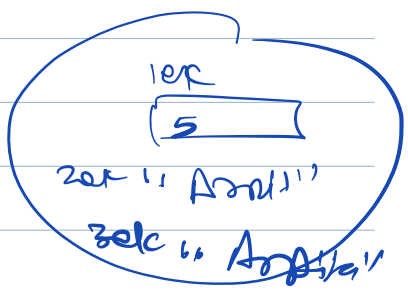
```
str += 'a'
```



```
gun ( ) {
    String str = "Arpit"
    int [] arr = {10, 20}
    fun (str, arr)
    print (str);
    print (arr[0]);
}
```







$(a \% m) + m$