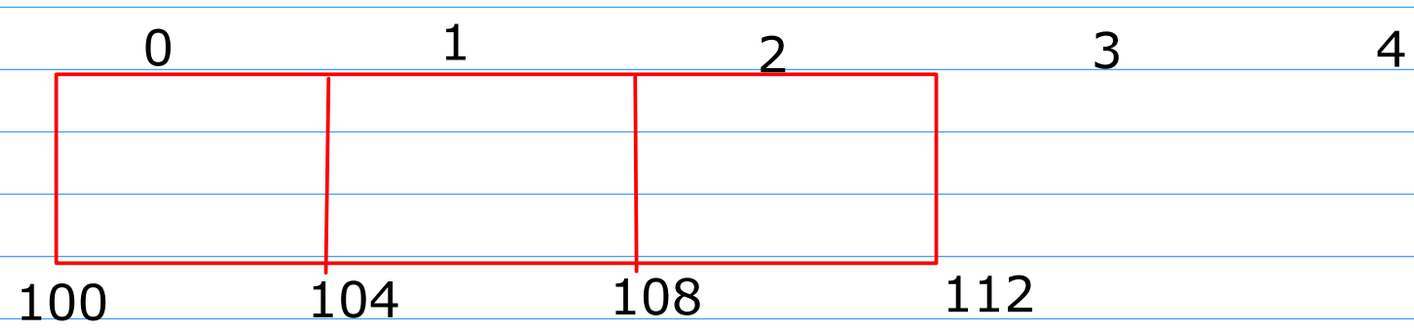
Array -> Reference Type

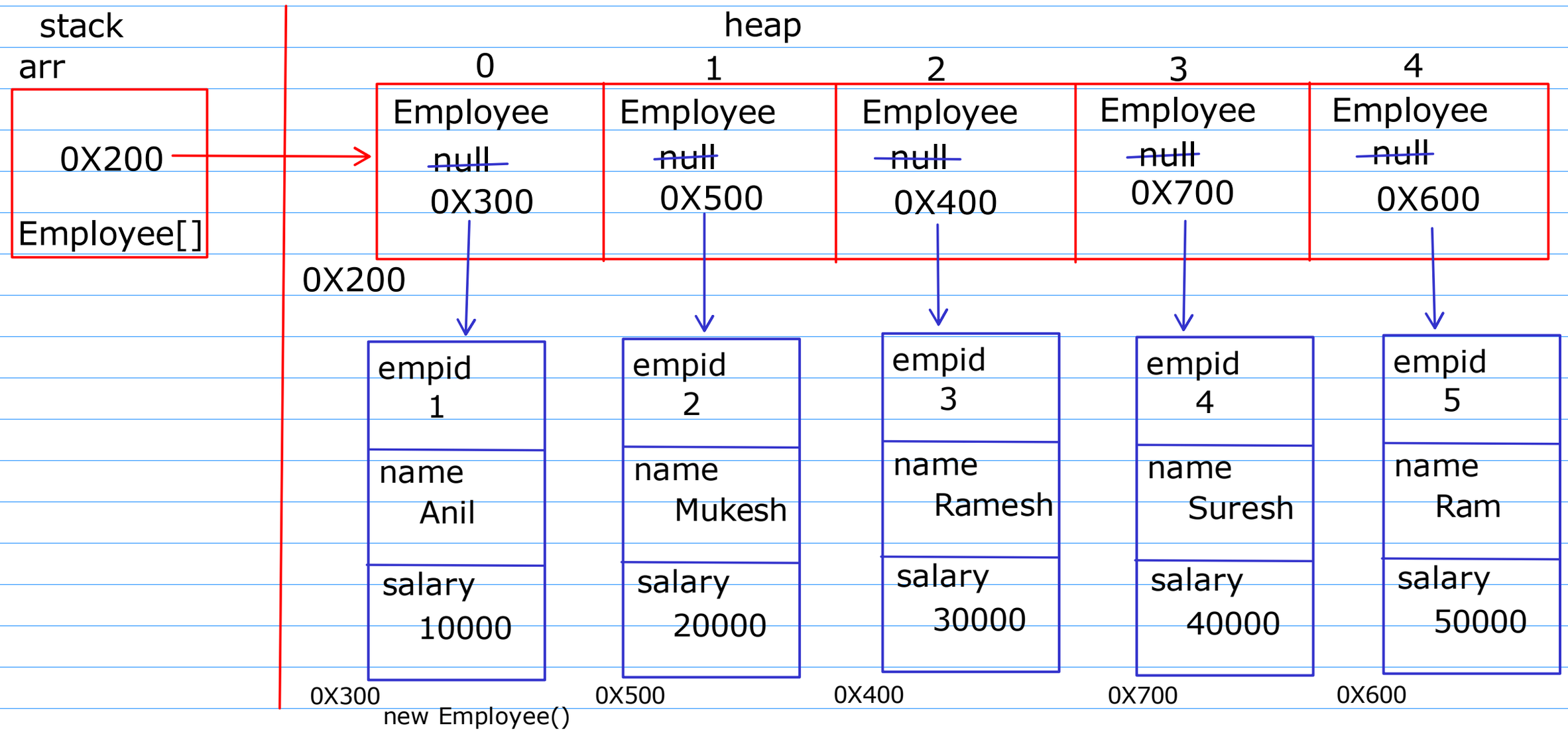
1. Single Dimensional Array
2. MultiDimensional Array
3. Ragged Array

```
int []arr = new int[5];
int [][] arr = new int[2][3];
int [][] arr = new int[2][];
arr[0] = new int[3];
arr[1] = new int[2];
```

[][]

```
for(int[] ele:arr)
```

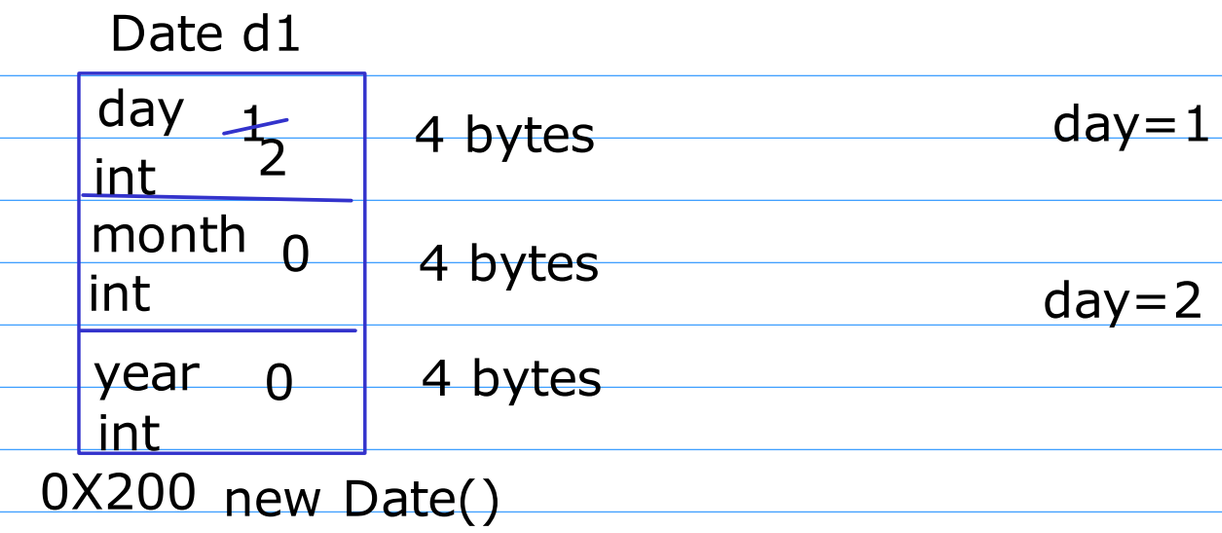




Practice Homework

```
Student[][] arr = new  Student[5][];
 arr[0] = new Student[5]; // DAC
 arr[1] = new Student[3];//DMC
 arr[2] = new Student[2];// DBDA
```

```
for(:)
// can we initialize the array
```



Static

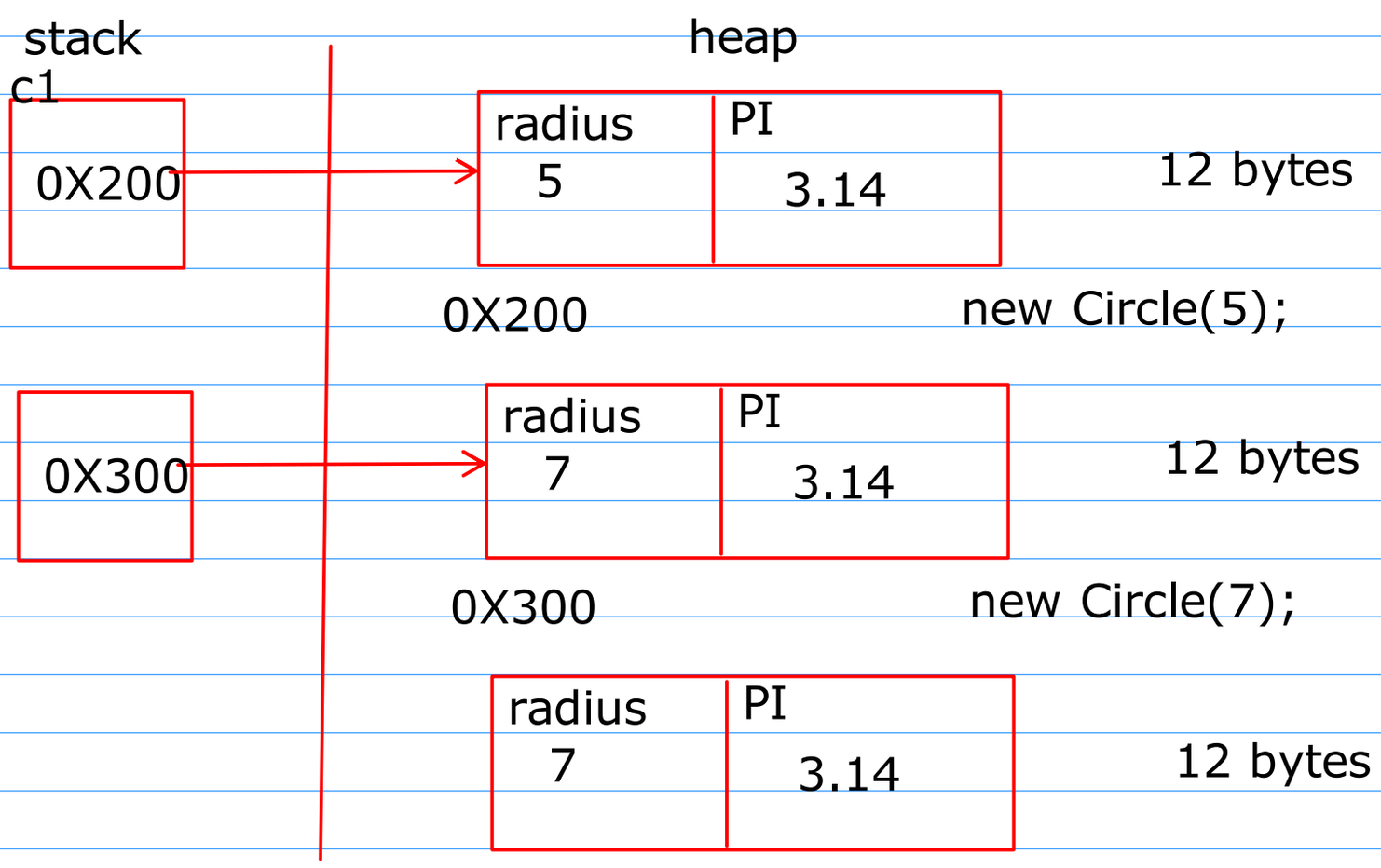
- Sharing

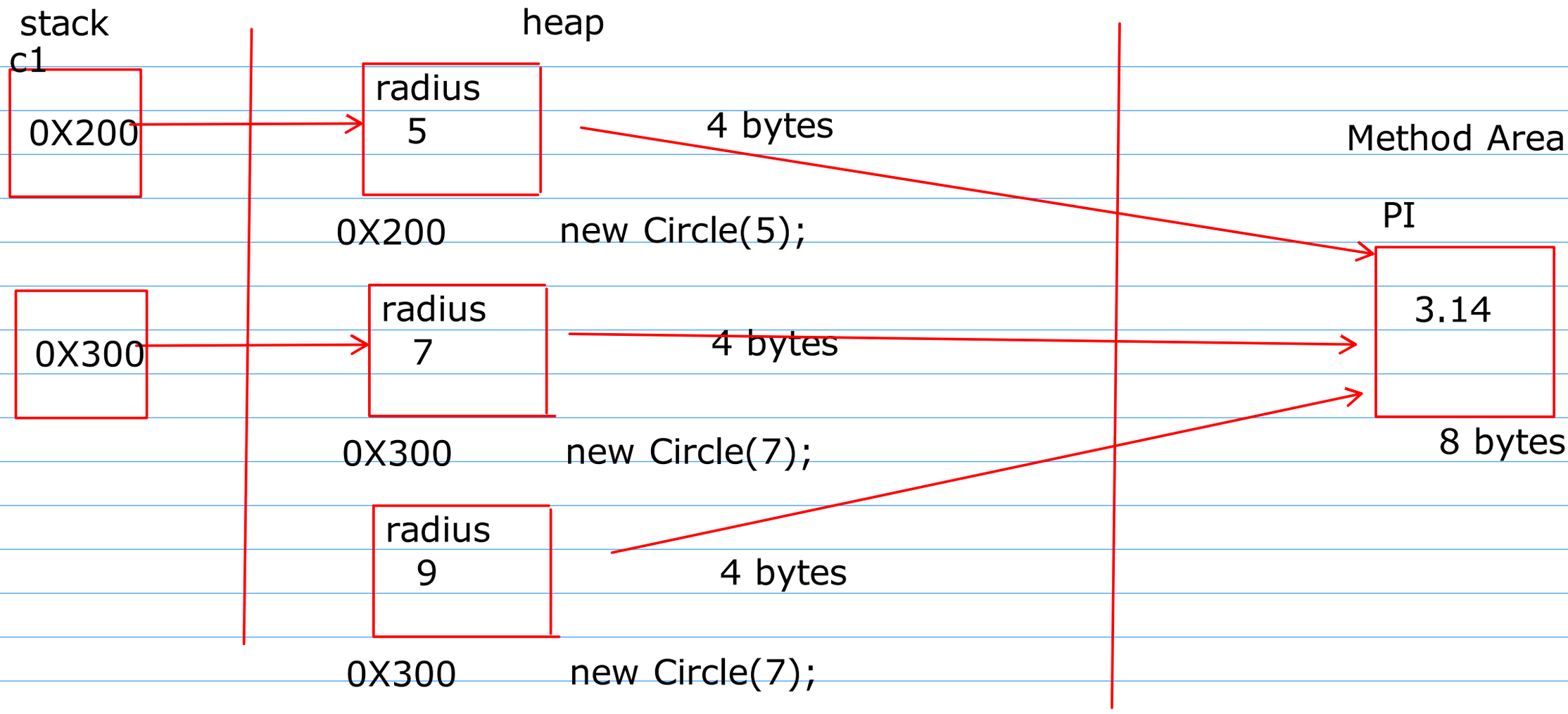
Class

- Consists of Fields and methods
- The fields and methods can be static as well as non static

Object

- The size of an object is equal to size of all the non static fields of the class
- Only the non static fields gets the memory inside the object
- Static fields do not get the memory in the object





Static Fields
- we can declare the fields of the class as static
- static fields do not get the memory inside the object
- these fields gets the memory on the method area only once at the time of class loading

```
int num1 = 1;
print(num1); //1
num1 = 2;
print(num1); // 2
num1 = num4;
print(num1);// 1
```

num4 = 1

```
class Test{
int num1;
static int num2;
}
```

Test t1 = new Test();
only when object of class is created then and only then the memeory for the non static field is provided.
non static fields will get the memory as many times as the objects are created
non static fields gets the memory on heap

static fields gets the memory even before creating any objects
static fields get the memory on the method area

```
class Program{
public static void main(){
}
}
```

```
Program p1 = new Program();
p1.main();

Program.main();
```

Program.main()

```
javac Program.java
java Program
```

```
String s1 = "10";

int n1 = Integer.parseInt(s1);
```

Static
- Fields
- Methods

Singleton Design Pattern
- To create only a single instance of a class