Java class Members

Elements declared inside the class body are known as class members.
There are two types of class members.

- Static class members.
- Non-static class members.

The function which are declared inside the class body are called as member functions.

Variables which are declared inside the class body are known as member variables.

The class members declared using static keyword are known as static members.

The member variables declared using static keyword are called as static member variables.

The member functions declared using static keyword are known as static member functions.

Static members are loaded to the memory during class loading and will be loaded only once into the memory. Static member should be accessed using class name.

Syntax:
Classname.static_member_name.

The class members declared without static keyword are known as non-static members.

The member variables declared without static keyword are called as non-static member variables.

The member functions declared without static keyword are known as non-static member functions.

Non-static members are loaded to the memory during object creation and will be loaded into memory whenever objects are created for a class.

Java class templates

How to declare a class?
class class_name ⟶ Class declaration
{
static datatype variable_name; ⟶ static variable
datatype variable_name; ⟶ non static variable

static returntype function_name()
{
⟶ Static member function
}
returntype function_name()
{
⟶ Non-static member function
}
}

In a source file we can declare multiple classes, but in order to execute we have to call class function containing main function.

```
class Java_class2
{

public static void main(String args[])
{
System.out.println("Demo function");
}

}
class sample
{

}

class test
{

}
```

Static members are accessed outside class using calssname.membername.

```
class Demo
{
static int a=10;
static int b=20;

static void test()
{
System.out.println("Static method is running");
}
}

class Java_class3
{
public static void main(String args[])
{
System.out.println("the value of a is"+ Demo.a);
Demo.test();
}
}
```

In java we can declare two types of variables:

1. Primitive type
2. Non-primitive type

Primitive type variable

Variable declared using data type is called primitive data variable

Syntax:
Datatype name_variable

Eg:
int a;

Non-Primitive type variable

Variable declared using the class type is known as non-primitive type variable.

Syntax:
Classname variablename

Eg:
Demo d;

For non-primitive type in right hand side either we can assign null value or object.

Syntax to create an object

Classname variablename= new constructor();

Non-primitive or object reference variable.

// declaring and calling non-static members

```
class Demo
{
int a=10;

void test()
{
System.out.println("non-Static method is running");
}
}
class Java_class4d
{

public static void main(String args[])
{

Demo d=new Demo();

System.out.println("the value of a is"+ d.a);
d.test();
}
}
```

Static members can be accessed directly both inside static context and non-static context within same class.
Eg:

```
class Java_class5
{
static int a=10;
static void test()
{
System.out.println("non-Static method is running");
}
public static void main(String args[])
{

System.out.println("the value of a is"+ a);
test();
}
}
```

Non-static members can be accessed directly inside a non-static context of same class

```
class Java_class6
{

int a=10;

void test()
{
System.out.println("the value of a is"+ a); \\ directly calling variable
}
public static void main(String args[])
{
Java_class6 j=new Java_class6();

j.test();
}
}
```

Non-static mebers cannot be accessed directly inside static context, it must be accessed using an object.

```
class Java_class6
{
int a=10;
void test()
{
System.out.println("the value of a is"+ a); \\ directly calling variable
}
public static void main(String args[])
{
Java_class6 j=new Java_class6();
/*object is created and called within a static method for a non-static method and variable*/
j.test();
System.out.println("the value of a is"+ j.a);
}
}
```

Write difference between static and non-static members:

Static members	**Non-static members**
1. Declared with static keyword	1. Declared without static keyword
2. Variables are loaded only once into memory	2. Are loaded whenever an object is created.
3. Accessed from different class giving classname.member	3. Accessed from different class by creating an object and reffering via that object

Local variables are neither static nor non-static hence should not declared local variable as a static variable.

```
Class demo
{
Static int a;
Static void test()
{
Static int b=20; //error
}
}
```

Create a class declare both static and non static members and access the members

```
class Java_class1
{

static int a=10;
double b=10.1;


static void add()
{
System.out.println("the static method");
}

void sub()
{
System.out.println("The non static method");
}

public static void main(String args[])
{

Java_class1 obj= new Java_class1();

System.out.println("The value of  a is"+a);
System.out.println(" The value of b is"+obj.b);

add();

obj.sub();

}

}
```

Calling method/variable	**How to call**
1. Static members within different classes.	1. Calling by name of the class Eg: demo a;
2. Non-Static members within different classes.	2. Calling by creation of object Eg: demo c=new demo() c.a;
3. Static members within same class, Non-static and static context.	3. Calling members directly.
4. non-Static members within same class and non-static context.	4. Calling members directly.
5. non-Static members within same class and static context	5. Calling by creation of object Eg: demo c=new demo() c.a;