JA111

Day-4

What is difference between assignment and initialization?

Constructor

- A constructor has name same as of its class name.
- It should be defined inside the class hence It can access instance member directly without using object.
- It is called automatically after object is created, before new operator completes.
- A constructor does not have an return type because implicit return type of a constructor is object of class type itself.
- A constructor is used for initialization and resource allocation to be done at the time object creation.

```
class-name(parameter-list){
  //body of constructor
}
```

- Constructor with no parameter: default constructor
- Constructor with parameter: parameterized constructor
- Can a class has multiple constructors? Yes

Data Type	Default Value
byte	0
short	0
int	0
long	0
float	0.0f
double	0.0d
char	'\u0000'
References	Null
boolean	False

The this keyword

- It is a reference variable that points to the calling object of method & constructor
- It is maintained by the system, you cannot perform assignment operation on this keyword

Application of this keyword

- To access the hidden instance variables
- To call constructor from another constructor

Why do we need static?

static variable

Only a single copy of the static variable is maintained and this copy is shared among all the objects

The static variable does not belong to object actually it belong to the class. It is also referred as class variable.

Because the static variable is for class, so to access the static variable, class-name can be used. Although accessing using the object name is also okay. Both (class-name and object name) will access the same variable

```
class-name.variable-name; (Preferred way)
obj-name.variable-name;
```

java does not allow the local static variable

Memory allocation for such variables happens only once when the class is loaded in the memory.

local variables: stack; dynamic variable (i.e. object and array): heap; static variable: metaspace (before JDK 1.8 the space was PermGen)

Tip: Metaspace is a memory area where static variables are created

static Method

- To create a static method, we have to precede the method prototype (i.e. return-type method-name(para-list)) with static keyword
- The static method belongs to the class, not to the object. So it can be called using class-name as well as using object-name. Both will call the same method

```
class-name.method-name(argument-list); (Preferred way)
object-name.method-name(argument-list);
```

- Because the static method belongs to the class so this keyword (and super keyword also) is not accessible in the static method and due to absence of this keyword, non-static (i.e. instance variables + methods) are not accessible in the static context (i.e. method or block)
- The non-static method (i.e. instance methods) can access the static variable directly because they are for class and can be accessed using the class name and the object name
- The static method of a class can access all static variables and methods of the same class directly.
- If programmer is creating local object/taking the object as parameter then it can access the instance variable using that local object. (because this keyword will not be involved)

static Block

- The static block of a class is executed first.
- It is executed even before main() method.

```
static{
  body of static block
}
```

The has-a relationship (Composition/Aggregation)

- It means that an instance of the one class has a reference to the instance of another class or the other instance of the same class.
- This relationship helps to minimize the duplication of code as well as the bugs.