25/07/2022

Java Notes Day3

Day 3
1. OOPS concepts - Inheritance
2. Constructor
3. Function overloading, Function overriding
4. Passing object as argument and returning object

Function overloading
It is a compiler activity(compile time activity)
Functions can be defined with the same name, differentiated by
-type of parameter
-number of parameters
-sequence of type of parameters
-Order of calling does not matter

Operator overloading

```
      -Sysout(10 + 30 +" sum is "+10+20)
          int     |  int |  |------> String
           30    String.valueOf(30)
                 |
                 "30"
                 |
                 String.concat("30")
                 |
                 "30 sum is"
```

Inheritance
-Derivation vs Implementation
-Derivation
Properties are acquired by the class
-Implementation
Object class is the root class of the java hierarchy
There are 11 methods in it -
toString, equals, hashCode - sysout(obj1) - prints hashcode,
clone, finalize, wait, wait, wait, notify, notifyAll, getClass
sysout(obj1) - Executes the hashcode method - toString converts the code into string and prints
Object class ---> class A (super() in constructor) ---> extends to inherit properties
Person ----> Student ----> Employee
super() - It is used to call a constructor and always called from the first line of a constructor
super. - It is used to call a method, function or data from the super class

----

Constructor
Has the same name as that of enclosing class
It is a function

Does not have return type, but it can take parameters, can be overloaded
Used to initialize an object - Object is only called once
Can have either implicit or explicit(multiple) declaration only - not both

```
WHAT  It is a function that has the same name as that of enclosing class
WHEN  Invoked during object creation process
WHY   To initialize the object
HOW   class A
      {
            A()
            {

            }
      }
```

```
Object - State + Behavior
       - Data  + Functionalities
```

Overriding

Redefining a function with the same name - in the child class from the parent class

----

```
Stack                  Heap
                        Field area                        Method area
p1                   gender name age               Constructor(),
toString()
100                                           |           |

                                              |           |
p2                gender name age class grade      Constructor(),
toString()
200                                           |           |
                                              |           |
p3          gender name age class grade empId salary     Constructor(),
toString()
300
```

Passing object as a parameter to a method

? When to use the class name and when to use object reference

Returning object as a return value from a method

Same name of the variable is possible when variables are maintained by different methods