Name: Gautam K K

Department: Testing- Impact Trainee

Task: JAVA

JAVA

Primitive Datatypes

Datatypes specify how memory stores the variable

In java, primitive datatypes are not part of any class or object, they are predefined to store of values of specific type

Datatypes:

- Byte = 1byte – 8bit upto 128
- Char = 2byte – 16bit (store one letter or ascii value)
- Boolean = 1bit – techically not precise
- Short = 2byte – 16bit
- Int = 4byte – 32bit
- Long = 8byte – 64bit
- Double = 8byte – 64bit (16 digit)
- Float = 4byte – 32 bit (7 digit)

Reference/Object type:

- Non –primitive datatypes are used to represent the objects
- They are called as reference type ->because in the memory of that object it stores the reference address where the value is present
- String is the non primitive datatype, Arrays, Classes
- Reference type datatypes are refer to instances or objects INTEGER number= 10; means integer is the class and the number is the object of that class
- Non-primitive datatype is user defined – means the class is there in built-in but we can create any methods, feilds and so on with that, which is not predefined
- Likewise the array of Student object which is not defined ex: int[] regno = new int();

- similarly Student[] student = new Student[]; - userdefined

Wrapper class

- A wrapper class is the class that is used to convert the primitive datatypes to Object and viceversa
- For example if we want to store it in the collection, collection only allow objects type since collection is designed to store and manipulate the group of objects, so that it only allows object type
- Then we use wrapper class -> since the feilds or data members in the object may contain the primitive dt if we wnt to manipulate o access that dt we want wrapper class to convert
- Each primitive dt ==> has its Wrapper class
- The process of converting the primitive --> object is called autoboxing
- The process of converting the primitive --> object is called Unboxing
- And it also has the methods like Integer.parseInt() to convert
- Generics in java is the type safety we give to collecion like arraylist<Integer> which can store only this type --> in that Objects only allowed not primitives

Naming Standard

- Varibles –use camelcase
- Method – use camelcase, method's action (verb)
- Class – Pascalcase, (noun)
- Constants – UPPERCASE like (final int REG_NO = 12379;)
- Boolean – can be used with is, has,can,should
- Abstract class – with pascal case or upper camelcase
- Interfaces – With pascal case and verb or action ;

Lifecycle of thread

1. Creation – new thread(), rthread.start()
2. Runnable – ready to run and waiting for cpu to allocate time
3. Blocked – synchornized (){} - when the thread try to access which is already with other thread
4. Waiting – thread.wait() ->will wait untill it got thread.notify() from other thread, thread.join(), thread.sleep()
5. Terminated – if it finishes the exceution it will terminated or if we invoke the thread.interrupt() it terminate --> tocheck we can use thread.isalive()

Composition and Aggregation

Aggregation

- Aggregation is **has a** relationship between the parent and child
- In Aggregation, the child object can exist without the parent object, this implies the loosely associated objects

Composition

- Composition has **part-of** relationship between objects
- In composition if the parent object is destroyed then the child object also destroyed, then it is strongly coupled between the object