```
WrapDemo.java:10: warning: [removal] Byte(byte) in Byte has been deprecated and
marked for removal
      Byte g1 = new Byte(grade); // wrapping

JDK9 constructor usage is deprecated use "valueOf"

can we write our own method in runnable interface or only run() is allowed in
runnable interface?
   inbuilt class/interface can't be modified, we need to just take the benift.

if any excepiton occur in one thead is there any effect on other thread in program?
      MulitThreading -> Best suited only when tasks are indepedent of each other
                                It is choose to improve the application performance
by using CPU time effectively.
      main() --------> t1---------> t2
     Thread => seperate stack for execution
     3 Threads -> 3 seperate Stacks

if main method is present in program,that means automatically thread is created for
main thread,or
we need to manually create a thread for main method?
sir can you please explain why hder sir said that main thread is not named after
main i did
not understood that conecpt??
default thread creation happens first ? or bringing the method to the runtime stack
area first ? could you clarify pls
            main() is loaded from method Area to stack and thread will start the
execution.
            JVM will create one thread with that thread it starts the execution.
            When the execution starts the stack should be given for the Thread.
            Inside stack the body of main method is available so the stack name is
"main".

sir in single run() example little confusion came, all the threads created and
linked....
but there is if-else condition ...so once one thread get executed then only others
will get chance ?
  Ans.  totally depends on ThreadScheduler Algorithm(part of JVM)

At what time thread go to dead state : for example t1.join(); -> for this when will
t1 will be in dead state?
      once the complete execution of t1 is done only then Thread will enter into
dead state.

Life cycle of thread is same as "Life cycle of Human Being".
     new/born ----------------> Ready/runnable -----run()-----> running
state--------finsished with execution---------> dead state

Q>
is multithreding with sigle run method used in real time or in industry level
projects?
MultiThreading -> Many tasks
        1 Thread --> 1 Task

can duck the exception in run method when thread.sleep method is used in run
method...?
     interface Runnable{
            public void run();
     }
```

Q>at first point of entry jvm will look forward for public static void method and later invoke
main thread without main method no java application will start to work is it correct?
yes,it results in Exception.

Q> Can write our own custom Scheduler ?
We can write but for this sequence we don't have any "SRS"(interface)

in thread class constructor what all the lines will be there?
Thread class constructor will have a call to Object class constructor.

Q>
sir at the starting when hyder sir print thread name,priority and method name,it was printing
main,5 and main,how 5 cames as a priority for main thread?
JVM will give defautl priorirty for the main thread.
Default priority is "5".

ContextSwitching => Switching the control from one thread to another thread by ThreadScheduler is called "ContextSwitching".
q1. why is it considered that threads are faster in contect switching also?
ContextSwitching in mulithreading is done by JVM(program) compared to ContextSwitching done by OS
since os is not involved we say ContextSwitching is faster at threads level.
q2. what does, Threads use shared memory area mean?
In mulitThreading application jvm will maintain the stack region for sepearte threads and these regions
data can be interchanged b/w threads, so we say Threads works in Shared memory.

if run method was called inside start method then what is the difference between calling direct run method and start method?
Thread is a class present in "java.lang.Thread"
Thread class start method
1. Register Thread with ThreadScheduler
2. It performs all low level memory activities(usage of shared memory)
3. It makes a call to run()

```
Thread t1=new Thread();
    t1.run();
```

sir please explain, is it possible to start thread two times, & can we directly call the
run() instead start(), multiple time this question asked to me while i am giving interview.

```
    Thread t1=new Thread();
      t1.start();
      t1.start();//IllegalThreadStateException
```

Q>

```
Integer i=12;//AutoBoxing using valueOf()
System.out.println(i);//i.toString() will be called so "12" in String format.
System.out.println(i+7);//19(Integer Object) will be printed.
```

Q>

```
Integer i=new Integer("20");
```

```
Integer i2=new Integer("21");
System.out.println(i2);//calls i2.tostring() to print the data
System.out.println(i+i2);// System.out.println(20+21);//41 will be printed.
```

If 3 threads are in runnable state and if thread scheduler has given control to say

thread 1 and if it's not going to sleep or wait or blocked state in between will it

complete it's execution first or in between will thread 2 or 3 will begin it's execution

Thread-1 -> If it is not entered into sleeping state/waiting state then it will complete the execution

Then the T.S will give control to other thread(T2,T3)depends of Algorithm.

can we start a thread again, Sir?

Ans.No

why does for the same application, let's say just incrementing variable by one each time by multiple threads,

we get different outputs each time? why is it so

Ans. Becoz of Concurrency.

How ThreadScheduler decide which thread to help chance ? And is ThreadScheduler behave like queue ?

It is not in the hands of the programmer,totaly depends on the vendor algorithm.

byte bb=12;//compiler will treat as "int", but the datatype u supplied is byte where the value is with in the range.
Byte b=Byte.valueOf(bb);
short s=153;//compiler will treat as "int", but the datatype u supplied is short where the value is with in the range of short
Short S=Short.valueOf(s);
Sir,Above Examples working fine but In below example, why we can't to pass value directly in valueOf() method

```
Byte b1=Byte.valueOf(11);//compile time
Short S1=Short.valueOf(44);//compile time
```

Q>
Sir, We have main thread to execute. In the main method consider we have two threads. To enter the two

threads in running state we need to have a delay in the currently running thread?

```
main(){
            Thread t1=new Thread();
            t1.start();//create a sepearte statck for 't1' and inform Scheduler to schedule the Thread.

            Thread t2=new Thread();
            t2.start();
}
```

need one example for converting primitive to stringtype

```
String s= Integer.toString(10);
System.out.println(s);
```

Q1.****if run method was called inside start method then what is the difference

between calling direct run method and start method?******

Q2.if main thread has default priority as 5 then wt will have 1,2,3,4 and why 5 will execute first instead of 1,2,3,4 and
wt are those whose has priority as 1,2,3,4?

```
public static void main(String[] args) {
            Demo d = new Demo();
            d.start();
            System.out.println(d.getName());
            System.out.println(d.getPriority());
               System.out.println(d.getState());  //TIMED_WAITING what it means?
      }
```

Actually singly thread is being executed in runningstate then how we are calling it as multiple task at a time
overall application -> multiThreading
JVM perspective -> it is single threading only

if we have created a t1 thread inside main and t1 got exception then entire remaining execution part of main thread also stops?

```
class Demo extends Thread{
      public void run() {
                  int c= 10/0;
      }
}
  class Test{
            public static void main(String[] args){

                        Thread t =new Thread(new Demo());
                        t.start();

                        System.out.println("hello");//prints becoz seperate
stack(Thread scheduler control)

            }
  }
```

output: depends on T.S(if main thread is given a chance then hello) otherwise "Abnormal termination".