NPTEL Online Certification Courses

Indian Institute of Technology Kharagpur

PROGRAMMING IN JAVA

Assignment6

TYPE OF QUESTION: MCQ

Number of questions: 10 **Total mark: 10 × 1 = 10**

QUESTION 1:

Which of the following is NOT a method of the Thread class in Java?

a. public void run()
b. public void exit()
c. public void start()
d. public final int getPriority()

Correct Answer: b

Detailed Solution:

The java.lang.System.exit() method terminates the currently running Java Virtual Machine. It is a status code where a nonzero or 1 indicates abnormal termination of the program whereas zero or 0 indicates normal termination of the program. It is not included in the thread class as it is not the part of the execution cycle of the method.

QUESTION 2:

Which of the following statement is true in case of starting a thread with 'run()' and 'start()' method?

a. There is no difference between **starting a thread with 'run()' and 'start()' method.**
b. When you call **start()** method, main thread internally calls **run()** method to start newly created Thread
c. When you call **run() method** directly no new **Thread** is created and code inside **run()** will execute on current **Thread**.
d. None

Correct Answer: b

Explaination:

Main **difference** is that when program calls **start() method** a new **Thread** is created and code inside **run() method** is executed in new **Thread** while if you call **run() method** directly no new **Thread** is created and code inside **run()** will execute on current **Thread**.

QUESTION 3:

Which of the following can be used to create an instance of Thread?

a. By implementing the Runnable interface.
b. By extending the Thread class.
c. By creating a new class named Thread and calling method run().
d. By importing the Thread class from package.

Correct Answer: a, b

Detailed Solution:

An application that creates an instance of Thread must provide the code that will run in that thread. There are two ways to do this:

- *Provide a Runnable object.* The Runnable interface defines a single method, run, meant to contain the code executed in the thread. The Runnable object is passed to the Thread constructor
- *Subclass Thread.* The Thread class itself implements Runnable, though its run method does nothing. An application can subclass Thread, providing its own implementation of run

Reference:https://docs.oracle.com/javase/tutorial/essential/concurrency/runthread.html

---

QUESTION 4:

What is the output of the following program?

```
public class Question
    {
     public static void main(String[] args) {

          try {
              int a=5/0;
             System.out.print("a ");
          } catch (ArithmeticException ae) {

              System.out.print("ArithmeticException ");
              }catch (Exception e) {
                 System.out.print(" Exception ");
          }
            System.out.print("Hello World");
      }
}
```

a. Hello World
b. ArithmeticException
c. ArithmeticException Exception Hello World
d. ArithmeticException Hello World
e. none

Correct Answer: d

Detailed Solution: ArithmeticException is an unchecked exception in Java that occurs due to an exceptional arithmetic condition. This generally indicates that a mathematical error has occurred at run-time which can't be dealt with, for example, when an integer is divided by zero.

Since here “devide by zero” is already catched by “Arithmatic exception” so catch “Exception” block will not execute.

QUESTION 5:

Which method restarts a dead thread

a. start()
b. restart()
c. restartThread()
d. none

Correct Answer: d

Detailed Solution: Thread can not be restarted you have to create a new Thread everytime.
A thread is born, started, runs, and then dies.Once a thread enters dead state it cannot be restarted.

QUESTION 6:

Assume the following method is properly synchronized and called from a thread A on an object B: wait(2000); After calling this method, when will the thread A become a candidate to get another turn at the CPU?

a. After thread A is notified, or after two seconds.
b. Two seconds after thread A is notified.
c. After the lock on B is released, or after two seconds.
d. Two seconds after lock B is released.

Answer : **a**

Explanation: Either of the two events (notification or wait time expiration) will make the thread become a candidate for running again

---

QUESTION 7:

The following is a simple program using the concept of thread.

```
public class Question extends Thread{
 public void run(){
  for(int i=1;i<5;i++){

    System.out.println(++i);
  }
 }
 public static void main(String args[]){
    Question t1=new Question();
    t1.run();
```

What is the output of the above program?

a. 1
   3
b. 1
   2
   3
   4
c. Runtime error
d. 2
   4

Correct Answer: d

Detailed Solution:

QUESTION 8:

For the program given below, what will be the output after its execution?

```
public class Main{
public static void main(String[]args){
        Thread thread=Thread.currentThread();
        System.out.print(thread.activeCount());
        thread.run();
        System.out.print(thread.activeCount());
       }
}
```

a. 01
b. False True
c. True True
d. 11

Correct Answer: d

Detailed Solution:

java.lang.Thread.activeCount() : Returns an estimate of the number of active threads in the current thread's thread group and its subgroups.

QUESTION 9:

Which of the following is/are not a correct constructor for a thread object?

a. Thread(Runnable a, String str);
b. Thread(Runnable a, int priority);
c. Thread(Runnable a, ThreadGroup t);
d. Thread(int priority);

Correct Answer: b,c,d

Detailed Solution:
Thread(Runnable a, String str)creates a new Thread object. The others are not valid constructors to create a thread object.

QUESTION 10:

Which exception is thrown when an array element is accessed beyond the array size?

a. ArrayElementOutOfBounds
b. ArrayIndexOutOfBoundsException
c. ArrayIndexOutOfBounds
d. None of these

Answer: B

Explanation: *ArrayIndexOutOfBoundsException* is thrown when an array element is accessed beyond the array size.

**************END**************