PROGRAMMING IN JAVA

Assignment 6

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. `isAlive()`
b. `getPriority()`
c. `getNames()`
d. `sleep()`

Correct Answer: c

Detailed Solution:

getName() is a method in the pre-defined Java class Thread but not getNames(). Other methods like isAlive(), getPriority() and sleep() are defined in the Thread class.

QUESTION 2:

Which of the following method can be used to know the priority of a thread?

a. `getPriority()`
b. `priority()`
c. `isRunning()`
d. `getThreadPriority()`

Correct Answer: a

Detailed Solution:

getPriority() is the method, which is used to know the priority given to a thred.

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:

A thread is better defined as

a. a basic unit of CPU utilization.
b. a control used to manage multiple requests by the same user without having to have multiple copies of the program.
c. a hardware device like Processor.
d. a multiple copies of the same program.

Correct Answer: a, b

Detailed Solution:

A thread is a basic unit of CPU utilization. Multi-threading is an execution model that allows a single process to have multiple code segments (i.e., threads) running concurrently within the “context” of that process. Multi-threading is the ability of a process to manage its use by more than one user at a time and to manage multiple requests by the same user without having to have multiple copies of the program.

QUESTION 5:

Which of the following statement is NOT true about a thread?

a. A piece of code that runs concurrently with other threads.
b. A statically ordered sequence of instructions.
c. Used to express concurrency on both single and multiprocessor machines.
d. A thread can be executed independently.

Correct Answer: d

Detailed Solution:
Threads are not independent of one another like processes. And, as a result threads share with other threads their code section, data section, and OS resources (like open files and signals).

QUESTION 6:

Which of the following will contain the body of the thread?

a. `run();`
b. `start();`
c. `stop();`
d. `main();`

Correct Answer: a

Detailed Solution:
The `run()` method of a thread is same as the `main()` method for an application. Starting the thread causes the object's run method to be called**.**

QUESTION 7:

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

```
public class Question7 extends Thread{
      public void run(){
            System.out.println("Thread started …");
      }
      public static void main(String args[]){
            Question7 t1 = new Question7();
            t1.start();
      }
}
```

How many threads will be there when the above program is in execution?

a. 0
b. 1
c. 2
d. 3

Correct Answer: c

Detailed Solution:
The main thread and t1 thread altogether count to 2 threads.

QUESTION 8:

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

```
public class Question8{
    public static void main(String[] args) {
        Thread thread = Thread.currentThread();
        System.out.println(thread.isAlive());
    }
}
```

a. 0
b. true
c. 1
d. false

Correct Answer: b

Detailed Solution:
isAlive() returns a boolean value depending on whether a thread is alive or not.

QUESTION 9:

Which of the following is 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: a

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:

What `notifyAll()` method does?

a. Wakes up all threads that are waiting on this object's monitor.
b. Wakes up only one thread among a group of threads that are waiting on this object's monitor.
c. Wakes up all threads that are not waiting on this object's monitor.
d. It doesn’t Wake up any thread that are waiting on this object's monitor.

Correct Answer: a

Detailed Solution:
notifyAll() : Wakes up all threads that are waiting on this object's monitor. A thread waits on an object's monitor by calling one of the wait methods.

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