1 **ESSAY**

Q1: Main Thread Control

Write a Java program to achieve the tasks below:

- Change the name of the main thread to " **SkyThread** ". (1pt)
- Change the priority of the current thread to **MAX_PRIORITY**. (1pt)
- Print the **name, priority, and group** of the current thread. (1pt)
- Print the **numbers from 1 to 5**, with a **delay of 2 seconds** between each number. (2pt)

*You must attach a **screenshot** of the code and output.*

2 **MULTIPLE CHOICE**

Which statement correctly obtains a **reference to the currently executing thread** in Java?

A - Thread t = new Thread();

B - Thread t = Thread.start();

C - Thread t = Thread.currentThread();

D - Thread t = currentThread();

3 **MULTIPLE CHOICE**

Why is the **Runnable** interface commonly used in multithreaded Java programs?

A - It automatically stops threads

B - It provides the *run()* method that defines the thread execution code

C - It replaces the *Thread* class completely

D - It increases CPU speed

4 **MULTIPLE CHOICE**

A **race condition** occurs when:

- A - Multiple threads access shared data at the same time without proper synchronization
- B - A thread waits until another thread finishes execution
- C - Threads execute one after another in an organized order
- D - Threads execute sequentially

5 **MULTIPLE CHOICE**

What is the purpose of **synchronization** in Java?

- [ ] A - To pause thread execution for a period of time
- [x] B - To protect shared data from race conditions
- [x] C - To allow only one thread to access shared data at a time
- [ ] D - To create multiple threads automatically

6 **MULTIPLE CHOICE**

Deadlock Problem:

Why may using *sleep()* inside a synchronized method cause deadlock?

- A - Because the thread releases the lock while sleeping
- B - Because the sleeping thread still holds the lock
- C - Because sleep() stops all threads permanently
- D - Because sleep() creates another synchronized method

7 **MULTIPLE CHOICE**

Avoiding Deadlock:
Which method allows a thread to wait and release the lock so other threads can continue execution?

A - *sleep()*

B - *run()*

C - *wait()*

D - *start()*