Institute of Computer Technology

B. Tech. Computer Science and Engineering

Semester: III

Sub: Object-Oriented Programming

Course Code: 2CSE303

Practical Number:13

Objective: To learn about threading concepts in Java.

Problem Definition:

**Q.1. What is the concept of threading in Java? Explain with the help of a sample program example.**

**Ans:-** Threading in Java allows the execution of multiple threads (small units of a process) concurrently. It enhances the efficiency of CPU utilization and allows multitasking.

E.g

class MyThread extends Thread {

public void run() {

System.out.println("Thread is running...");

}

}

public class ThreadingExample {

public static void main(String[] args) {

MyThread t = new MyThread();

t.start(); // Starts the thread

}

}

**Q.2. Explain single and multithreading in java with real life examples.**

**Ans:-** **Single-threading:** Only one task executes at a time. Example: Reading a file line by line.

**Multithreading:** Multiple tasks execute simultaneously. Example: Listening to music while downloading a file.

**Real-life Example:**

* Single-threading: A person cooking one dish at a time.
* Multithreading: A person cooking multiple dishes simultaneously.

**Q.3. Write an appropriate program by using the following method:**

1. **run() method.**

**Ans:-** class MyThread extends Thread {

public void run() {

System.out.println("Running thread...");

}

}

public class RunMethodExample {

public static void main(String[] args) {

MyThread t = new MyThread();

t.start();

}

}

1. **start() method.**

**Ans:-** class MyThread extends Thread {

public void run() {

System.out.println("Thread started!");

}

}

public class StartMethodExample {

public static void main(String[] args) {

MyThread t = new MyThread();

t.start();

}

}

1. **sleep() method.**

**Ans:-** class SleepExample extends Thread {

public void run() {

for (int i = 1; i <= 5; i++) {

try {

Thread.sleep(1000); // 1-second delay

} catch (InterruptedException e) {

System.out.println(e.getMessage());

}

System.out.println(i);

}

}

}

public class SleepMethodExample {

public static void main(String[] args) {

SleepExample t = new SleepExample();

t.start();

}

}

1. **yield() method.**

**Ans:-** class MyThread extends Thread {

public void run() {

for (int i = 0; i < 5; i++) {

Thread.yield();

System.out.println(Thread.currentThread().getName() + " is running");

}

}

}

public class YieldMethodExample {

public static void main(String[] args) {

MyThread t1 = new MyThread();

MyThread t2 = new MyThread();

t1.start();

t2.start();

}

}

1. **join() method.**

**Ans:-** class JoinExample extends Thread {

public void run() {

for (int i = 1; i <= 5; i++) {

try {

Thread.sleep(500);

} catch (InterruptedException e) {

System.out.println(e.getMessage());

}

System.out.println(i);

}

}

}

public class JoinMethodExample {

public static void main(String[] args) {

JoinExample t = new JoinExample();

t.start();

try {

t.join();

} catch (InterruptedException e) {

System.out.println(e.getMessage());

}

System.out.println("Thread finished!");

}

}

**Q.4. Write an appropriate program by using the following method.**

1. **wait() method.**
2. **notify() method.**
3. **notifyAll() method.**

**E.g**

**Ans:-** class SharedResource {

synchronized void waitMethod() {

try {

System.out.println("Waiting...");

wait(); // Waits until notified

} catch (InterruptedException e) {

System.out.println(e.getMessage());

}

System.out.println("Resumed");

}

synchronized void notifyMethod() {

System.out.println("Notifying...");

notify(); // Wakes up a single thread

}

}

public class WaitNotifyExample {

public static void main(String[] args) {

SharedResource resource = new SharedResource();

Thread t1 = new Thread(resource::waitMethod);

Thread t2 = new Thread(resource::notifyMethod);

t1.start();

try { Thread.sleep(1000); } catch (InterruptedException e) {}

t2.start();

}

}

**Q.5. Write an appropriate program by using the following method.**

**1. isAlive() method.**

**2. suspend() method.**

**E.g**

**Ans:-** class MyThread extends Thread {

public void run() {

System.out.println("Thread is running...");

}

}

public class IsAliveExample {

public static void main(String[] args) {

MyThread t = new MyThread();

t.start();

System.out.println("Is thread alive? " + t.isAlive());

}

}