Assignment 14

Multithreading

1. **What do you mean by multithreading and ? why it is important ?**
   Ans: 1. Executing several thread at a time is call multithreading.
   Thread: separate flow of execution is called as ”thread”.

   In the concept of multithreading there are executing several task at a time . In case of multithreading these all task are the part of same program.
   why it is important ?

- To implement multimedia graphics
- To develop web application server
- To develop game

To working with multithreading , java developer need to write only 10% of code remaining code is take care by jvm.

2. **What are the benefits of using multithreading ?**
   - **Ans:** To working with multithreading , java developer need to write only 10% of code remaining code is take care by jvm.
   - Multithreading allows the execution of multiple parts of a program at the same time.
   - These parts are known as threads and are lightweight processes available within the process. So multithreading leads to maximum utilization of the CPU by multitasking.

3. **What is thread in java?**
   Ans: Thread:separate flow of execution is called as ”thread”.
   If there is only one thread then it is called as single thread program**.**
   For ever thread ther would be a separate job.
   We can define thred in two way….
   1. By implementing Runnable interface
   2. By extending thread class

4. **What are the two for implementing thread in java ?**
   Ans:
   1. By implementing Runnable interface
   2. By extending thread class

5. **What are the difference e between process and thread ?**

   Ans: A process is an instance of a program that is being executed or processed.

   Thread is a segment of a process or a lightweight process that is managed by the scheduler independently.

   Processes are independent of each other and hence don't share a memory or other resources.

   Threads are interdependent and share memory.

6. **How can we create daemon thread ?**

   Ans: In Java, daemon threads are low-priority threads that run in the background to perform tasks such as garbage collection or provide services to user threads. The life of a daemon thread depends on the mercy of user threads, meaning that when all user threads finish their execution, the Java Virtual Machine (JVM) automatically terminates the daemon thread.

```
// Java program to demonstrate the usage of
// setDaemon() and isDaemon() method.

public class DaemonThread extends Thread
{
  public DaemonThread(String name){
        super(name);
  }

  public void run()
  {
        // Checking whether the thread is Daemon or not
        if(Thread.currentThread().isDaemon())
        {
              System.out.println(getName() + " is Daemon thread");
        }

        else
        {
              System.out.println(getName() + " is User thread");
        }
  }

  public static void main(String[] args)
```

```
    {

        DaemonThread t1 = new DaemonThread("t1");
        DaemonThread t2 = new DaemonThread("t2");
        DaemonThread t3 = new DaemonThread("t3");

        // Setting user thread t1 to Daemon
        t1.setDaemon(true);

        // starting first 2 threads
        t1.start();
        t2.start();

        // Setting user thread t3 to Daemon
        t3.setDaemon(true);
        t3.start();
    }
}
```

Methods of Daemon Thread

1. void setDaemon(boolean status):

This method marks the current thread as a daemon thread or user thread. Setting a user thread as a daemon can be done using the**'tU.setDaemon(true)'**, while setting a daemon thread as a user thread can be done using the **'tD.setDaemon(false)'**.

Syntax:

public final void setDaemon(boolean on)

Parameters:

- **on:** If true, marks this thread as a daemon thread.

7. What are the slepp() and wait() method in java ?

Ans: wait(): immediately thread will enter into the waiting state.

Sleep(): if a thread doesn't want to perform any operation for a particular of time then we should do to sleep().

Every sleep method throe Interruption error , ehich can be handled by try and catch block.