Assignment 14

1. What do you mean by Multithreading? Why is it important?

Ans: Executing several tasks simultaneously where each task is a separate independent part of the same process is called as multithreading. It is important feature in java because it saves the cpu time and makes the execution fast.

2. What are the benefits of using Multithreading?

Ans: Improved performance through parallel execution.

- Responsive user interfaces by offloading tasks to separate threads.
- Concurrency for efficient resource utilization.
- Parallelism for simultaneous execution of independent tasks.
- Task separation and better code organization.
- Efficient resource sharing and communication.
- Scalability to handle more tasks or clients concurrently.
- Enhanced resource management.
- Real-time processing capabilities.

3. What is Thread in Java?

Ans: A thread is very light weighted process or we can say the smallest part of the process that allows program to operate more efficiently by running multiple tasks simultaneously.

4. What are the two ways of implementing thread in Java?

Ans: There are two ways to create a thread

1. By extending Thread class

Eg.
```
class Demo extends Thread
 {
public void run()
{
System.out.println("thread is started");
}
public static void main(String args[])
{
Demo obj=new Demo();
obj.start();
}
}
```

2. By implementing Runnable interface

Eg.
```
class Demo implements Runnable
 {
public void run()
{
System.out.println("thread by runnable interface");
}
public static void main(String args[])
{
Demo obj=new Demo();
Thread obj1 =new Thread(obj);
obj1.start();
}
}
```

5. What's the difference between thread and process?

Ans: Thread: It refers to the smallest unit of the process. It is used to execute different parts of the program at the same time.

Process: It refers to a program that is in execution i.e. an active program. A process can be handled using PCB (Process Control Block).

6. How can we create daemon threads?

Ans: We can create daemon threads in java using the thread class setDaemon(true). It is used to mark the current thread as daemon thread or user thread. isDaemon() method is generally used to check whether the current thread is daemon or not. If the thread is a daemon, it will return true otherwise it returns false.
Example:

7. What are the wait() and sleep() methods?

Ans: In Java, the wait() method is used to pause the execution of a thread until another thread signals that it can resume. When a thread calls wait() on an object, it releases the lock on the object and waits until another thread calls notify() or notifyAll() on the same object.

sleep() method can be used to pause the execution of the current thread for a specified time in milliseconds. The argument value for milliseconds cannot be negative. Otherwise, it throws IllegalArgumentException