

Object-Oriented Programming (CS F213)

Labsheet-10

Prof.R Gururaj

Practice Problem-1

// Making main thread to wait till the child thread completes.

```

class MyThread extends Thread
{
    MyThread(String s)
    {
        super(s);
        start();
    }
    public void run()
    {
        System.out.println(Thread.currentThread()+" started execution:");
        try{
            for(int i=0;i<5;i++)
            {
                System.out.println(" CHILD"+ " Printing :"+i);
                Thread.sleep(500);
            }
        }
        catch(InterruptedException ie)
        {
            System.out.println(Thread.currentThread()+" Interrupted:");
        }
    }
}

class JoinDemo1
{
    public static void main(String args[])
    {
        Thread mt=new MyThread("Child "); // child thread
        Thread t=Thread.currentThread(); // getting reference to main thread
        t.setName("MAIN THREAD:");
        System.out.println(t.getName()+" started :");
        try{
            mt.join(); // main thread will wait till mt finishes
        }
        catch(InterruptedException ie)
        { System.out.println(mt+ " Interrupted:"); }
        System.out.println(" Last statement in Main thread :");
    }
}

```

Practice problem-2

// To demonstrate result of executing multiple Threads without synchronization

// Practice Problem-3

//To modify the code in (3) to make it Synchronized

Practice Problem-4

// Demonstrating the usage of ArrayList to store all sorts of objects

Practice Problem-5

// Demonstrating ArrayList to store specific (Integer) type objects

Practice Problem-6

// Demonstrating Hashtable/HashMap storing key-value pairs

Practice Problem-7

// Demonstrating Collections class to reverse and ArrayList and to use other algorithms

Practice Problem-8

// Demonstrating Stack

Practice Problem-9

// Demonstrating Vector. It is like ArrayList but its methods are synchronized

Other Important Classes in Util package

Random	Generate pseudo random numbers.
Observable	The Observable class is used to create subclasses that other parts of your program can observe. When an object of such a subclass undergoes a change, observing classes are notified. Observing classes must implement the Observer interface, which defines the

	update() method. The update() method is called when an observer is notified of a change in an observed object
Date	The Date class encapsulates the current date and time. This also implements the comparable interface
BitSet	A utility class to hold bit values as Boolean values in a special type of array
StringTokenizer	Text processing, to break a given string into constituent tokens.
Scanner	Used to read input from Console, File, String. Or any source that implements Readable interface

Exercise:

If time permits the instructor will give some problems to solve based the practice problems.
