Read Me:-

i. Before going through below exercises please visit the link given below, where you can experience the coding standard that each and every developer should follow.

ii. This Code Conventions for the Java Programming Language document contains the standard conventions that Sun follow and recommend that we should follow. It covers filenames, file organization, indentation, comments, declarations, statements, white space, naming conventions, programming practices and includes a code example.

iii. LINK - http://www.oracle.com/technetwork/java/codeconv-138413.html

Multithreading Lab Exercise Day-1
Duration: 2 Hours

Program: 1

Write a program to create a class 'ThreadDemo' which extends Thread class and having some instance variables as following ,

```
private String firstName;
private String secondName;
private long aWhile;
```

ThreadDemoclass implements a parameterized constructor as following to initialize the above instance variables.

```
public ThreadDemo (String firstName, String secondName, long delay) {
this.firstName = firstName;
this.secondName=secondName;
While = delay;
setDaemon(true);
}
```

This ThreadDemo class also overriding the run() method as following

```
public void run() {
try{
while(true) {
System.out.print(firstName);
sleep(aWhile);
System.out.print(secondName + "\n");
}
} catch (InterruptedException e) {
System.out.println(firstName + secondName + e);
}
}
```

write a program to create a class ‘ThreadDemoTest’ which implements main method as per the following structure.

```
public static void main(String[] args) {
//Create instances of ThreadDemo class by passing appropriate parameters to their parameterized constructors.
Thread first = new ThreadDemo("A ", "a ", 200L);
Thread second = new ThreadDemo("B ", "b ", 300L);
Thread third = new ThreadDemo("C ", "c ", 500L);
System.out.println("Press Enter when you have had enough...\n"); //Start the threads . .
first.start();
second.start();
third.start(); try {
System.in.read(); System.out.println("Enter pressed...\n");
} catch (IOException e) {
//Display the Exception in the console.
}
}
```

Note: In the above program try to implement Runnable interface instead of extending Thread class.

Program: 2

Write a program to create a class ‘NewThread’ which implements Runnable interface having a instance variable ‘t’ of thread type. This class ‘NewThread’ implements a constructor as following

```
NewThread() {
//create a thread object by passing appropriate parameters and assign it to instance variable.
t = newThread(this, "Demo Thread");
//Display the thread in the console with some appropriate message.
System.out.println("Child thread: " + t);
t.start(); // Start the thread
}
```

The class ‘NewThread’ also override the run method as per the following structure.

```
public void run() {
try {
for(inti = 5; i > 0; i--) {
System.out.println("Child Thread: " + i);
Thread.sleep(500);
}
} catch (InterruptedException e) {
    System.out.println("Child interrupted.");
}
```

```
    System.out.println("Exiting child thread.");
}
```
Write a program to create a class ‘MainThread’ which implements main method as per the following structure.

```
public static void main(String args[]) {
new NewThread();
try{
for(inti = 5; i > 0; i--) {
System.out.println("Main Thread: " + i);
Thread.sleep(1000);
}
} catch (InterruptedException e) {
  System.out.println("Main thread interrupted.");
}
   System.out.println("Main thread exiting.");
}
```
Note : In the above program extend Thread class instead of implementing Runnable Interface.

Program: 3 Get current thread ..

Write a program to create a class ‘CurrentThreadDemo’ which extends java.lang.Thread class and override the run() method as follows

```
for (inti = 0; i < 5; i++) {
printMyName();
}
```
Where printMyName() method is implemented as following . .

```
public void printMyName() {
System.out.println("The Thread name is " + Thread.currentThread().getName());
}
```
Now implement the main method as following

```
public static void main(String[] args) {
```
//Create an object ‘ttsn’ of ‘currentThreadDemo’ here.

//Now set a new name as ’created one’ to the above created thread using setName() method of java.lang.Thread class.

//start the thread using start() method of java.lang.Thread class.

```
Thread t2 = currentThread();//Create a thread object using currentThread() method.
```
//Set a name as ‘main one’ to the newly created thread using setName() method of java.lang.Thread class.

```
for(inti = 0; i < 5; i++) {
```
//Print the name of this newly created thread object usiong the above method printMyName().

```
ttsn.printMyName();
}
```

}

Exepected output :

The Thread name is Main One The Thread name is Main One The Thread name is Main One The Thread name is Main One The Thread name is Main One The Thread name is Created One The Thread name is Created One The Thread name is Created One The Thread name is Created One The Thread name is Created One

All the Best