**Hands On Lab 1**

**[Extending Thread class](https://personales.unican.es/corcuerp/java/Labs/LAB_23.htm" \l "Exercise_1)**

A thread is a thread of execution in a program. The Java Virtual Machine allows an application to have multiple threads of execution running concurrently.  This hands-on lab takes you through the basics of using Java threading.

## **Exercise 1: Extending Thread class**

In this exercise, you are going to learn how to create and start a thread execution by writing a class that extends Thread class.  You will learn how to start the thread by either not having the start() method in the constructor of the subclass or having it in the constructor of the subclass.

1. **[The start() method is NOT in the constructor of the subclass](https://personales.unican.es/corcuerp/java/Labs/LAB_23.htm" \l "1.1)**
2. **[The start() method is in the constructor of the subclass](https://personales.unican.es/corcuerp/java/Labs/LAB_23.htm" \l "1.2)**

### **(1.1) The start() method is NOT in the constructor of the subclass**

0. Start Spring Tools Suite IDE if you have not done so yet.  
1. Create a new Java project

* Select **File->New Java Project (Alt+Shift+N)**. The **New Java Project** dialog box appears.
* For the Project Name field, type in **ExtendThreadClassTest0**as project name.
* Choose ‘Use default JRE (11….). Make sure you have set your default JRE set to version 11 and the appropriate compliance settings as well. Deselect ‘Create module-info.java file’. Click Next.Click Finish.
* Observe that **ExtendThreadClassTest0**project appears
* Right click on the project and select **File**->**New Class.**
* Enter in a package name of your choosing or use the default. Type in **ExtendThreadClassTest0 a**s the class name. The main method stub should also be created.
* The IDE generated **ExtendThreadClassTest0.java** is displayed in the source editor window of STS IDE.

2. Modify the IDE generated **ExtendThreadClassTest0.java**as shown in Code-1.11 below.  Study the code by paying special attention to the bold fonted parts.  Note that the **start()** is invoked after the object instance of **PrintNameThread** class is created.

|  |
| --- |
| public class ExtendThreadClassTest0 {         public static void main(String args[]) {                **// Create object instance of a class that is subclass of Thread class**         System.out.println("Creating PrintNameThread object instance..");         **PrintNameThread pnt1 =                 new PrintNameThread("A");**               **// Start the thread by invoking start() method**         System.out.println("Calling start() method of " + pnt1.getName() + " thread");         **pnt1.start();**             } } |

Code-1.11: ExtendThreadClassTest0.java  
  
3. Write **PrintNameThread.java** as shown in Code-1.12 below.

|  |
| --- |
| **// Subclass extends Thread class public class PrintNameThread extends Thread** {         PrintNameThread(String name) {         super(name);     }      **// Override the run() method of the Thread class.     // This method gets executed when start() method     // is invoked.     public void run() {         System.out.println("run() method of the " + this.getName() + " thread is called" );                 for (int i = 0; i < 10; i++) {             System.out.print(this.getName());         }**     } } |

Code-1.12: PrintNameThread.java  
  
4. Build and run the project

* Right click **ExtendThreadClassTest0**project and select **Run**.
* Observe the result in the **Output**window. (Figure-1.13 below)

|  |
| --- |
| Creating PrintNameThread object instance.. Calling start() method of A thread run() method of the A thread is called AAAAAAAAAA |

Figure-1.13: Result of running ExtendThreadClassTest0 application  
  
  
5. Modify the **ExtendThreadClassTest0.java**as shown in Code-1.15 below. The code fragments that need to be added are highlighted in **bold and blue-colored** font.

|  |
| --- |
| public class ExtendThreadClassTest0 {         public static void main(String args[]) {                  // Create object instance of a class that is subclass of Thread class         System.out.println("Creating PrintNameThread object instance..");         PrintNameThread pnt1 =                 new PrintNameThread("A");                 // Start the thread by invoking start() method         System.out.println("Calling start() method of " + pnt1.getName() + " thread");         pnt1.start();      **System.out.println("Creating PrintNameThread object instance..");         PrintNameThread pnt2 =                 new PrintNameThread("B");         System.out.println("Calling start() method of " + pnt2.getName() + " thread");         pnt2.start();                 System.out.println("Creating PrintNameThread object instance..");         PrintNameThread pnt3 =                 new PrintNameThread("C");         System.out.println("Calling start() method of " + pnt3.getName() + " thread");         pnt3.start();**     } } |

Code-1.15: Modified ExtendThreadClassTest0.java  
  
6. Build and run the project

* Right click **ExtendThreadClassTest0**project and select **Run**.
* Observe the result in the **Output**window. (Figure-1.16 below)

|  |
| --- |
| Creating PrintNameThread object instance.. Calling start() method of A thread Creating PrintNameThread object instance.. Calling start() method of B thread AAAAAAAAAACreating PrintNameThread object instance.. BCalling start() method of C thread BBBBBBBBBCCCCCCCCCC |

Figure-1.16: Result  
  
7. For your own exercise, modify **ExtendThreadClassTest0.java** as following. Build and run the application.

* Create and start another thread.
* Set the name of the thread as "MyOwn"

### **(1.2) The start() method is in the constructor of the subclass**

1. Create a new Java project

* Select **File**->**New Project (Ctrl+Shift+N)**. The **New Project** dialog box appears.
* For the Project Name field, type in **ExtendThreadClassTest2** as project name.
* Choose ‘Use default JRE (11….). Make sure you have set your default JRE set to version 11 and the appropriate compliance settings as well. Deselect ‘Create module-info.java file’. Click Next.Click Finish.
* Observe that **ExtendThreadClassTest2** project appears
* Right click on the project and select **File**->**New Class.**
* Enter in a package name of your choosing or use the default. Type in **ExtendThreadClassTest2 a**s the class name. The main method stub should also be created.
* The IDE generated **ExtendThreadClassTest2.java** is displayed in the source editor window of STS IDE.

2. Modify the IDE generated **ExtendThreadClassTest2.java**as shown in Code-1.21 below.

|  |
| --- |
| public class ExtendThreadClassTest2 {         public static void main(String args[]) {                         PrintNameThread pnt1 =                 new PrintNameThread("A");                         PrintNameThread pnt2 =                 new PrintNameThread("B");                         PrintNameThread pnt3 =                 new PrintNameThread("C");            } } |

Code-1.21: ExtendThreadClassTest2.java  
  
3. Write **PrintNameThread.java** as shown in Code-1.22 below.  Note that the **start**() method is invoked as part of the constructor method of the **PrintNameThread**class.

|  |
| --- |
| public class PrintNameThread extends Thread {         PrintNameThread(String name) {         super(name);  **// start() method is inside the constructor of the subclass         start();**     }         public void run() {         String name = getName();         for (int i = 0; i < 10; i++) {             System.out.print(name);         }     } } |

Code-1.22: PrintNameThread.java  
  
4. Build and run the project

* Right click **ExtendThreadClassTest2**project and select **Run**.
* Observe the result in the **Output**window. (Figure-1.23 below)

|  |
| --- |
| AAAAAAAAAABBBBBBBBBBCCCCCCCCCC |

Figure-1.23: Result of running ExtendThreadClassTest2 application  
  
5. For your own exercise, modify **ExtendThreadClassTest2.java** as following. Build and run the application.

* Create and start another thread.
* Set the name of the thread as "MyOwn"

### **Summary**

In this exercise, you have learned how to create and start a thread by extending Thread class.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*