

# Lab1: An Introduction to Java ME



## Objective

The objective to this lab is to:

1. Install the Sun Java Wireless Toolkit
2. Get you started with developing a simple Java ME (J2ME) application

## Experiment 1 Download and install the Sun Java Wireless Toolkit

In this experiment you will install the Sun Java Wireless Toolkit. This toolkit provides emulations, documentations and examples for developing Java applications targeted at CLDC/MIDP-compliant mobile phones.

To complete the experiment, do the following steps:

1. Minimum Hardware requirements are:
  - 100 MB hard disk space
  - 128 MB system RAM
  - 800 MHz Pentium III CPU
2. Install Java 2, Standard Edition (J2SE) Development Kit (SDK), version 1.5.0- if you plan to do development.  
To download the SDK go to <http://java.sun.com/javase/downloads/index.html>
3. Install Java 2, Micro Edition (J2ME) Wireless Toolkit( WTK)  
To download J2ME WTK on Windows, go to <http://java.sun.com/products/sjwtoolkit/download.html>, **Download Sun Java Wireless Toolkit 2.5.2 for CLDC for Windows and Linux**
4. Run the Wireless Toolkit  
Run KToolbar from the Start menu.

## Experiment 2 Explore the Wireless Toolkit

In this part you will get to experiment with the Sun Java Wireless Toolkit which is a very simple IDE for building MIDlets.

To get started, do the following steps:

1. Run the Sun Java Wireless Toolkit 2.5.2 for CLDC from Start menu
2. Open sample project "Games" (File -> Open Project)
3. Change the device setting to MediaControlSkin
4. Click on Build and Run it
5. Select the TitlePuzzle application and Play the game [**3 marks**]

*Solve the puzzle and let the instructor know when you finish.*

Select another project/application and explore the Toolkit.

### **Exercise 1** Write a simple MIDlet

In this exercise you will write a simple MIDlet that displays a simple text message and a few commands.

To get started, do the following steps:

1. Copy and paste the following class into a text file and save it with an appropriate name like FirstMIDlet.java ( the file name should be the same as the class name)

#### **Listing 1- FirstMIDlet.java**

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

// A MIDlet displaying a simple text and a few commands.
public class FirstMIDlet extends MIDlet
    implements CommandListener {

    //The exit, info, and buy commands
    private Command exitCommand;
    private Command infoCommand;
    private Command buyCommand;

    //The display for this MIDlet
    private Display display;

    public FirstMIDlet() {
        display = Display.getDisplay(this);
        exitCommand = new Command("Exit", Command.SCREEN, 1);
        infoCommand = new Command("Info", Command.SCREEN, 2);
        buyCommand = new Command("Buy", Command.SCREEN, 2);
    }

    // Start the MIDlet by creating the TextBox and
    // associating the exit command and listener.
    public void startApp() {
        TextBox t = new TextBox("FirstMIDlet",
            "Welcome to MIDP Programming", 256, 0);
        t.addCommand(exitCommand);
        t.addCommand(infoCommand);
        t.addCommand(buyCommand);
        t.setCommandListener(this);
        display.setCurrent(t);
    }
}
```

```

// Pause is a no-op because there are no background
// activities or record stores to be closed.
public void pauseApp() { }

// Destroy must cleanup everything not handled
// by the garbage collector.
// In this case there is nothing to cleanup.
public void destroyApp(boolean unconditional) { }

// Respond to commands. Here we are only implementing
// the exit command. In the exit command, cleanup and
// notify that the MIDlet has been destroyed.
public void commandAction(Command c, Displayable s) {
    if (c == exitCommand) {
        destroyApp(false);
        notifyDestroyed();
    }
}
}
}

```

In order to understand the code, read the following explanation carefully.

Note the followings from Listing 1:

```

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

```

- We import the midlet and lcdui packages. The midlet package defines the MIDP, and the lcdui package provides user interface APIs for implementing user interfaces for MIDP applications.

```

public class FirstMIDlet extends MIDlet
    implements CommandListener {
}

```

- Each MIDlet must extend the MIDlet class which allows for the orderly starting, stopping, and cleanup of the MIDlet. Each MIDlet has the following methods:

```

public void startApp() {}
public void pauseApp() {}
public void destroyApp(boolean unconditional) {}

```

- In FirstMIDlet, the Command class is used to create commands. The command itself contains only information about a command. When a command is activated, an action occurs. This action is defined in a CommandListener associated with the screen. Let's look at one of the command statements from Listing 1:

```
Command infoCommand = new Command("Info",Command.SCREEN, 2);
```

- As you can see, a command contains three pieces of information: a label, a type, and a priority. The label (which is a string) is used for the visual representation of the command. The type of the command specifies its intent. Finally, the priority value describes the importance of this command relative to other commands on the screen. A priority value of 1 indicates the most important command, and higher priority values indicate commands of lesser importance. The following shows how to define an action for the command “exitCommand”.

```
public void commandAction(Command c, Displayable s) {  
    if (c == exitCommand) {  
        ...  
    }  
}
```

- Demo your work to the TA. **[4 marks]**
2. Open a new project by clicking on New Project...  
A dialog will be opened. Give an appropriate name to the Project Name like Lab1\_p1. In the MIDlet Class Name, enter the class name, FirstMIDlet
  3. Copy the java class into the src folder of the project created. The project is created in WTK directory > apps sub-directory.
  4. Run the project in WTK22 environment. **[3 marks]**