Department of Information Technology

Mobile-Application-Development

Lab Manual

(IV B.tech -I Sem)

UDAY KIRAN.M
Asst. Professor

J.B.Institute of Engineering & Technology
Yenkapally, Moinabad(Mandal)
Himathnagar(post),Hydreabad

INDEX

Week - 1: Installation of Java Wireless Toolkit (J2ME)

1) If the Java Development Kit (JDK) is not there or only having the Java Runtime Environment (JRE) installed, install the latest JDK from http://java.sun.com/javase/downloads/index.jsp. Current stable release of Java is JDK 6 Update 7 but check the web page in case there are newer non-beta releases available.

2) Next, download the **Java Wireless Toolkit** (formerly called J2ME Wireless Toolkit) from: http://java.sun.com/products/sjwtoolkit/download.html.

3) Run the installer (for example, for Windows it is: sun_java_wireless_toolkit- 2_5_2-windows.exe). The installer checks whether a compatible Java environment has been pre-installed. If not, it is necessary to uninstall old versions of Java and perform Step 1 again.

Once after successful installation of Java and the tool kit compile this program and run the following program in the toolkit.

Steps to run this program in toolkit:

1. Start -> All Programs -> Sun Java Tool Kit -> Wireless Tool Kit
2. Click New Project – Enter Project Name -> Enter Class Name -> Click on Create Project.
3. Choose appropriate API Selection and Configurations.
4. Place Java Source file in WTK2.1 / WTK2.2\apps\projectname\src
5. Build the Project.
6. Run the Project.

```
import javax.microedition.lcdui.*; import javax.microedition.midlet.*;
public class HelloWorld extends MIDlet{ private Form form; private Display display;
public HelloWorld(){ super(); }
public void startApp(){ form = new Form("Hello World"); String msg = "Hello World!!!!!!!";
form.append(msg); display = Display.getDisplay(this); display.setCurrent(form); }
public void pauseApp(){} public void destroyApp(boolean unconditional){ notifyDestroyed();
} }
```

Printing Hello World program in J2ME

Printing Hello World program in J2ME

Step-1:-Start ->AllPrograms->Sun Java Tool Kit->Wireless Tool Kit

Step-2:-Click New Project –Enter project Name as FirstMidlet -> Enter ClassName as HelloMidlet->click on Create Project

Step-3:- A setting window will open up. Accept the defaults by clicking ok in that window.

Step-4:-Write Following Code in Notepad and save it as HelloMidlet.java

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class HelloMidlet extends MIDlet {
public HelloMidlet() {
}
public void startApp() {
Form form = new Form( "First Program" );
form.append( "Hello World" );
Display.getDisplay(this).setCurrent( form );
}
public void pauseApp() {
}
public void destroyApp( boolean unconditional ) {
}
}
```

Step-5:-Place HelloMidlet.java in C:\Documents and settings\ADMIN\j2mewtk\2.5.2\apps\FirstMidlet\src\

Step-6 :In the ktoolbar main window click on the “Build” button. When the build compiles successfully then click on the “Run” button.

Output:

Week - 2 Working with J2ME Features:

Working with J2ME Features: Say, creating a *Hello World* program Experiment with the most basic features and mobile application interaction concepts (lists, text boxes, buttons, radio boxes, soft buttons, graphics, etc)

2.1 Create a program which creates to following kind of menu.

* cut

* copy

* past

* delete

* select all

* unselect all

Create a program which creates a select menu

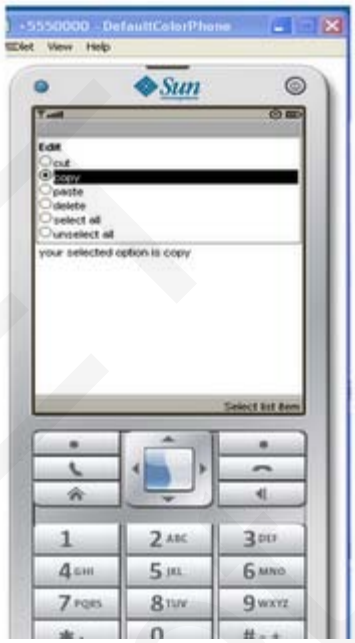
Aim: Develop a MIDlet Application for select list item

Source code:

```
 import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class MenuCreation extends MIDlet   implements CommandListener {
    public ChoiceGroup ch;
    public Form form;
    public Display display;
    public Command command;
     public StringItem st;
    public MenuCreation()
    {
        display=Display.getDisplay(this);
        ch=new ChoiceGroup("Edit",Choice.EXCLUSIVE);
        ch.append("cut",null);
        ch.append("copy",null);
        ch.append("paste",null);
        ch.append("delete",null);
        ch.append("select all",null);
        ch.append("unselect all",null);
        ch.setSelectedIndex(1, true);
        command=new Command("Select list item",Command.OK,1);
        form=new Form("");
        form.append(ch);
        form.addCommand(command);
        form.setCommandListener(this);
        st=new StringItem("","");
    }
    public void startApp() {
        display.setCurrent(form);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }

    public void commandAction(Command command,Displayable displayable)
    {
            if(command==command)
```

```
            {
                st.setText("");
                st.setText("your selected option is "+ch.getString(ch.getSelectedIndex()));
                form.append(st);
            }
    }
}
```

Output:



2.2 Event Handling.

Create a menu which has the following options:

* cut - can be on/off
* copy - can be on/off
* paste - can be on/off
* delete - can be on/off
* select all - put all 4 options on
* unselect all - put all 4 options off

Create a program which creates a select menu for Eventhandling

Event Handling.

Create a menu which has the following options:

* cut - can be on/off
* copy - can be on/off
* paste - can be on/off
* delete - can be on/off
* select all - put all 4 options on
* unselect all - put all 4 options off

Create a program which creates a select menu for Eventhandling
Aim: Develop a MIDlet Application for select menu for eventhandling
Source code:

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

public class MenuEvents extends MIDlet implements CommandListener,ItemStateListener {

    public ChoiceGroup ch;
    public ChoiceGroup ch1;
    public Form form;
    public Form form1;
    public Display display;
    public Command View;
    public Command Exit;
    public Command Back;
    public StringItem options;
    public Item item;
    public MenuEvents()
    {
        display=Display.getDisplay(this);
        form=new Form("");
        form1=new Form("Selcted Options are");
        ch=new ChoiceGroup("Preferences",Choice.MULTIPLE);
        ch.append("cut",null);
        ch.append("copy",null);
        ch.append("paste",null);
        ch.append("delete",null);
        ch.setSelectedIndex(1, true);
        form.append(ch);
        ch1=new ChoiceGroup("",Choice.EXCLUSIVE);
        ch1.append("select all",null);
        ch1.append("unselect all",null);
        ch1.setSelectedIndex(1, true);
        form.append(ch1);
        View=new Command("View",Command.OK,1);
        Exit =new Command("Exit",Command.EXIT,1);
        Back=new Command("Back",Command.BACK,1);
```

```
    form.addCommand(View);
    form.addCommand(Exit);
    form1.addCommand(Back);
    form.setCommandListener(this);
    form1.setCommandListener(this);
    form.setItemStateListener(this);
     }
  public void startApp()
       {
    display.setCurrent(form);
  }
  public void pauseApp() {
  }
  public void destroyApp(boolean unconditional) {
  }
  public void commandAction(Command command,Displayable displayable)
  {
        if(displayable==form)
        {
          if(command==View)
          {
             boolean opt[]=new boolean[ch.size()];
             options=new StringItem("","");
             String values="";
             ch.getSelectedFlags(opt);
             options.setText("");

             for(int i=0;i<opt.length;i++)< p=""> </opt.length;i++)<>
             {
                if(opt[i])
                {
                   values+=ch.getString(i)+"\n";
                                }
             }
             options.setText(values);
             form1.append(options);

          display.setCurrent(form1);
         }
else if(command==Exit)
          {

   destroyApp(true);
   notifyDestroyed();
}
    }
```

```
else if(displayable==form1)
          {
    if(command==Back)
    {
    display.setCurrent(form);
    options.setText("");
             }
}
  }
  public void itemStateChanged(Item item)
  {
     if(item==ch1)
     {
        int i=0;
         int size=ch.size();
         while(i<size)< p=""> </size)<>
         {
            if(ch1.getSelectedIndex()==0)
            ch.setSelectedIndex(i, true);
            else
               ch.setSelectedIndex(i, false);
            i++;
         }
     }
  }
}
```

Output:

2.3. Input checking
Create an MIDP application which examine, that a phone number, which a user has entered is in the given format.
* Area code should be one of the following: 040, 041, 050, 0400, 044
* There should 6-8 numbers in telephone number (+ area code)

Create a program which creates a select menu for Eventhandling

Event Handling.
Create a menu which has the following options:
* cut - can be on/off
* copy - can be on/off
* paste - can be on/off
* delete - can be on/off
* select all - put all 4 options on
* unselect all - put all 4 options off

Create a program which creates a select menu for Eventhandling
Aim: Develop a MIDlet Application for select menu for eventhandling
Source code:

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

public class MenuEvents extends MIDlet implements CommandListener,ItemStateListener {

    public ChoiceGroup ch;
    public ChoiceGroup ch1;
    public Form form;
    public Form form1;
    public Display display;
    public Command View;
    public Command Exit;
    public Command Back;
    public StringItem options;
    public Item item;
    public MenuEvents()
    {
        display=Display.getDisplay(this);
        form=new Form("");
        form1=new Form("Selcted Options are");
        ch=new ChoiceGroup("Preferences",Choice.MULTIPLE);
        ch.append("cut",null);
        ch.append("copy",null);
        ch.append("paste",null);
```

```
    ch.append("delete",null);
    ch.setSelectedIndex(1, true);
    form.append(ch);
    ch1=new ChoiceGroup("",Choice.EXCLUSIVE);
    ch1.append("select all",null);
    ch1.append("unselect all",null);
    ch1.setSelectedIndex(1, true);
    form.append(ch1);
    View=new Command("View",Command.OK,1);
    Exit =new Command("Exit",Command.EXIT,1);
    Back=new Command("Back",Command.BACK,1);
    form.addCommand(View);
    form.addCommand(Exit);
    form1.addCommand(Back);
    form.setCommandListener(this);
    form1.setCommandListener(this);
    form.setItemStateListener(this);
     }
public void startApp()
       {
    display.setCurrent(form);
}
public void pauseApp() {
}
public void destroyApp(boolean unconditional) {
}
public void commandAction(Command command,Displayable displayable)
{
       if(displayable==form)
       {
         if(command==View)
         {
             boolean opt[]=new boolean[ch.size()];
             options=new StringItem("","");
             String values="";
             ch.getSelectedFlags(opt);
             options.setText("");

             for(int i=0;i<opt.length;i++)< p=""> </opt.length;i++)<>
             {
                if(opt[i])
                {
                   values+=ch.getString(i)+"\n";
                                }
             }
             options.setText(values);
```

```
                form1.append(options);

              display.setCurrent(form1);
             }
else if(command==Exit)
              {

   destroyApp(true);
   notifyDestroyed();
}
      }
else if(displayable==form1)
          {
   if(command==Back)
   {
   display.setCurrent(form);
   options.setText("");
              }
}
  }
  public void itemStateChanged(Item item)
  {
     if(item==ch1)
     {
       int i=0;
        int size=ch.size();
        while(i<size)< p=""> </size)<>
        {
           if(ch1.getSelectedIndex()==0)
           ch.setSelectedIndex(i, true);
           else
              ch.setSelectedIndex(i, false);
           i++;
        }
     }
  }
}
```

Output:

Week - 3 Threads & High Level UI:

3.1. Create a slide show which has three slides, which includes only text. Program should change to the new slide after 5 seconds. After the third slide program returns to the first slide.

Create a slideshow which has three slides

Create a slideshow which has three slides.which includes only text.Program should change to the newslide after 5 seconds.After the third slide program returns to the First Slide
Source Code:

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class SlideShow extends MIDlet implements CommandListener {
   public Form slide1;
   public Form slide2;
   public Form slide3;
   public Command Exit;
   public Display display;
      public SlideShow()
   {
      display=Display.getDisplay(this);
      Exit=new Command("Exit",Command.EXIT,1);
      slide1=new Form("Slide1");
      slide1.append("This is  Slide number 1");
      slide1.addCommand(Exit);
      slide2=new Form("Slide2");
```

```
        slide2.append("This is  Slide number 2");
        slide2.addCommand(Exit);
        slide3=new Form("Slide3");
        slide3.append("This is  Slide number 3");
        slide3.addCommand(Exit);
        slide1.setCommandListener(this);
        slide2.setCommandListener(this);
        slide3.setCommandListener(this);
    }
   public void startApp() {
        Thread runner = new Thread(new ThreadRunner(display,slide1,slide2,slide3));
      runner.start();
    }
    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }
    public void commandAction(Command command,Displayable displayable)
    {
        if(displayable==slide1)
        {
            if(command==Exit)
                notifyDestroyed();
       }
 else if(displayable==slide2)
        {
            if(command==Exit)
                notifyDestroyed();
        }
        else if(displayable==slide3)
        {
            if(command==Exit)
                notifyDestroyed();
        }
    }
}
class ThreadRunner implements Runnable {
  Display display;
  public int c=0;
  public Form slide1;
  public Form slide2;
  public Form slide3;
   public  ThreadRunner(Display display,Form slide1,Form slide2,Form slide3) {
    this.display = display;
    this.slide1=slide1;
```

```
    this.slide2=slide2;
    this.slide3=slide3;
  }
  public void run() {
      while(true)
      {
      c++;
      if(c==1)
            display.setCurrent(slide1);
      else if(c==2)
         display.setCurrent(slide2);
      else if(c==3)
         display.setCurrent(slide3);
      else if(c==4)
         c=0;
try
      {
      Thread.sleep(1500);
         }
      catch(Exception ex)
      {
      }
      }
  }
}
```

Output:-

3.2 High-level UI
Create a MIDP application, which show to the user 5-10 quiz questions. All questions have 4 possible options and one right option exactly. Application counts and shows to the user how many right answers were right and shows them to user.

Create a MIDP application, which show to the user 5-10 quiz questions

Create a MIDP application, which show to the user 5-10 quiz questions. All questions have 4 possible options and one right option exactly. Application counts and shows to the user how many right answers were right and shows them to user
Source Code:

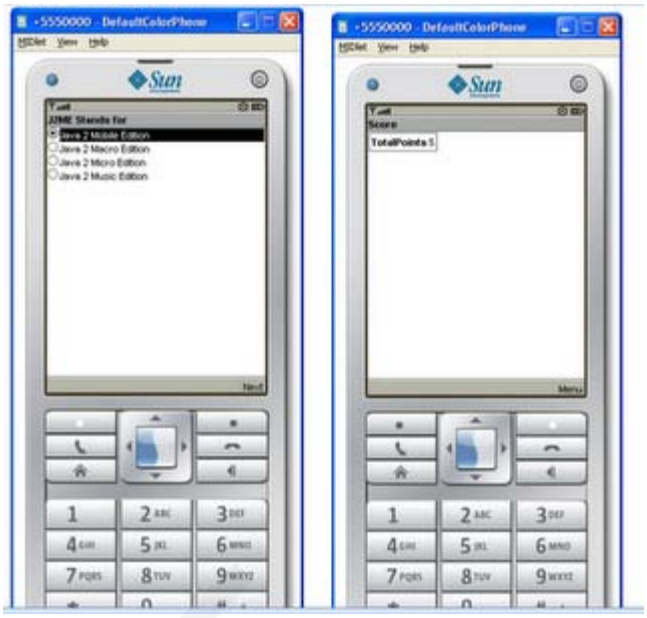
```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.rms.*;
import java.io.*;
public class Quiz extends MIDlet implements CommandListener {
    public Form form1;
   public Form form2;
   public Form form3;
   public Form form4;
   public Form form5;
   public Form form6;
   public Form form7;
   public ChoiceGroup ch1;
   public ChoiceGroup ch2;
   public ChoiceGroup ch3;;
   public ChoiceGroup ch4;;
   public ChoiceGroup ch5;;
   public Command nextCommand;
   public Command backCommand;
   public Command MenuCommand;
   public Command OkCommand;
   public Command ExitCommand;
   public Command sCommand;
   public Display display;
   public StringItem st;
   public TextField textfield;
   public int count;
   public RecordStore recordstore=null;
   public RecordEnumeration re=null;
   public Alert alert;
   public StringItem st1;
   public Quiz()
      {
      count=0;
```

```
    display=Display.getDisplay(this);
    nextCommand=new Command("Next",Command.OK,1);
    backCommand=new Command("Back",Command.BACK,1);
 st1=new StringItem("","");
textfield=new TextField("EnterName","",20,TextField.ANY);
form1=new Form("J2ME Stands for");
form2=new Form("a+b=");
form3=new Form("5*5");
form4=new Form("Who is AP CM");
form5=new Form("How many Districts in AP");
form6=new Form("Score");

ch1=new ChoiceGroup("",Choice.EXCLUSIVE);
ch1.append("Java 2 Mobile Edition", null);
ch1.append("Java 2 Macro Edition", null);
ch1.append("Java 2 Micro Edition", null);
ch1.append("Java 2 Music Edition", null);
form1.append(ch1);
form1.addCommand(nextCommand);
form1.setCommandListener(this);

ch2=new ChoiceGroup("",Choice.EXCLUSIVE);
ch2.append("b+a", null);
ch2.append("b*a", null);
ch2.append("b/a", null);
ch2.append("b-a", null);
form2.append(ch2);
form2.addCommand(nextCommand);
form2.addCommand(backCommand);
form2.setCommandListener(this);

ch3=new ChoiceGroup("",Choice.EXCLUSIVE);
ch3.append("20", null);
ch3.append("30", null);
ch3.append("10", null);
ch3.append("25", null);
form3.append(ch3);
form3.addCommand(nextCommand);
form3.addCommand(backCommand);
form3.setCommandListener(this);

ch4=new ChoiceGroup("",Choice.EXCLUSIVE);
ch4.append("Rosiah", null);
ch4.append("Jagan", null);
ch4.append("ChandaBabu", null);
ch4.append("Kiran", null);
```

```
    form4.append(ch4);
    form4.addCommand(nextCommand);
    form4.addCommand(backCommand);
    form4.setCommandListener(this);

    ch5=new ChoiceGroup("",Choice.EXCLUSIVE);
    ch5.append("8", null);
    ch5.append("4", null);
    ch5.append("11", null);
    ch5.append("23", null);
    form5.append(ch5);
    form5.addCommand(backCommand);
    form5.addCommand(nextCommand);
    form5.setCommandListener(this);
    form6.addCommand(ExitCommand);
    }
    public void startApp() {
       display.setCurrent(form1);
    }
    public void pauseApp() {
    }
    public void destroyApp(boolean unconditional) {
    }
    public void commandAction(Command cmd,Displayable displayable)
    {
 if(displayable==form1)
        {
            if(cmd==nextCommand)
            display.setCurrent(form2);
        }
 else if(displayable==form2)
        {
     if(cmd==nextCommand)
              display.setCurrent(form3);
     else if(cmd==backCommand)
     display.setCurrent(form1);
}
else if(displayable==form3)
        {
if(cmd==nextCommand)
display.setCurrent(form4);
else if(cmd==backCommand)
display.setCurrent(form2);
}
else if(displayable==form4)
{
```

```
if(cmd==nextCommand)
display.setCurrent(form5);
else if(cmd==backCommand)
display.setCurrent(form3);
}
else if(displayable==form5)
{
if(cmd==nextCommand)
{

if(ch1.getSelectedIndex()==2)
count++;
if(ch2.getSelectedIndex()==0)
            count++;
             if(ch3.getSelectedIndex()==3)
            count++;
            if(ch4.getSelectedIndex()==3)
            count++;
             if(ch5.getSelectedIndex()==3)
            count++;
             st.setText(String.valueOf(count));
                        form6.append(st);
            form6.append(textfield);
            display.setCurrent(form6);
            }
            }
        }
    }
```

Output:



Week - 4 Working on Drawing and Images

4.1 Create a slide show which has three slides, which includes pictures at PNG format. Program should change to the new slide other 5 seconds.

Create a slideshow which has three slides,which includes pictures at PNG format.Program should change to the new slide other 5 seconds

Create a slideshow which has three slides,which includes pictures at PNG format.Program should change to the new slide other 5 seconds.

Source Code:

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class imageSlideShow extends MIDlet implements CommandListener {
    public Form slide1;
    public Form slide2;
    public Form slide3;
    public Command Exit;
    public Display display;
    public Image image1;
    public Image image2;
    public Image image3;
    public ImageItem imageitem1;
    public ImageItem imageitem2;
    public ImageItem imageitem3;
       public imageSlideShow()
```

```
   {
display=Display.getDisplay(this);
          try
          {
             image1=Image.createImage("/1.png");
             image2=Image.createImage("/2.png");
             image3=Image.createImage("/3.png");
             imageitem1=new ImageItem(null,image1,ImageItem.LAYOUT_CENTER,"image1");
             imageitem2=new ImageItem(null,image2,ImageItem.LAYOUT_CENTER,"image2");
             imageitem3=new ImageItem(null,image3,ImageItem.LAYOUT_CENTER,"image3");
          }
          catch(Exception ex)
          {
          }
       Exit=new Command("Exit",Command.EXIT,1);
       slide1=new Form("Slide1");
       slide1.append(imageitem1);
       slide1.addCommand(Exit);
       slide2=new Form("Slide2");
       slide2.append(imageitem2);
       slide2.addCommand(Exit);
       slide3=new Form("Slide3");
       slide3.append(imageitem3);
       slide3.addCommand(Exit);
       slide1.setCommandListener(this);
       slide2.setCommandListener(this);
       slide3.setCommandListener(this);
   }
   public void startApp() {
      Thread runner = new Thread(new ThreadRunner(display,slide1,slide2,slide3));
     runner.start();
   }
   public void pauseApp() {
   }
   public void destroyApp(boolean unconditional) {
   }
   public void commandAction(Command command,Displayable displayable)
   {
      if(displayable==slide1)
      {
         if(command==Exit)
            notifyDestroyed();
      }
 else if(displayable==slide2)
      {
         if(command==Exit)
```

```
            notifyDestroyed();
        }
        else if(displayable==slide3)
        {
           if(command==Exit)
              notifyDestroyed();
        }
    }
}
class ThreadRunner implements Runnable {
  Display display;
  public int c=0;
  public Form slide1;
  public Form slide2;
  public Form slide3;
   public  ThreadRunner(Display display,Form slide1,Form slide2,Form slide3) {
    this.display = display;
    this.slide1=slide1;
    this.slide2=slide2;
    this.slide3=slide3;
  }
  public void run() {
      while(true)
      {
      c++;
      if(c==1)
            display.setCurrent(slide1);
      else if(c==2)
         display.setCurrent(slide2);
      else if(c==3)
         display.setCurrent(slide3);
      else if(c==4)
         c=0;
          try
      {
      Thread.sleep(1500);
          }
      catch(Exception ex)
      {
      }
      }
  }
  }
```

Output:-

4.2 Create a MIDP application, which draws a bar graph to the display. Data values can be given at int[] array.

Create a MIDP application,which draws a bargraph to display.Data values can be given at int[]array?

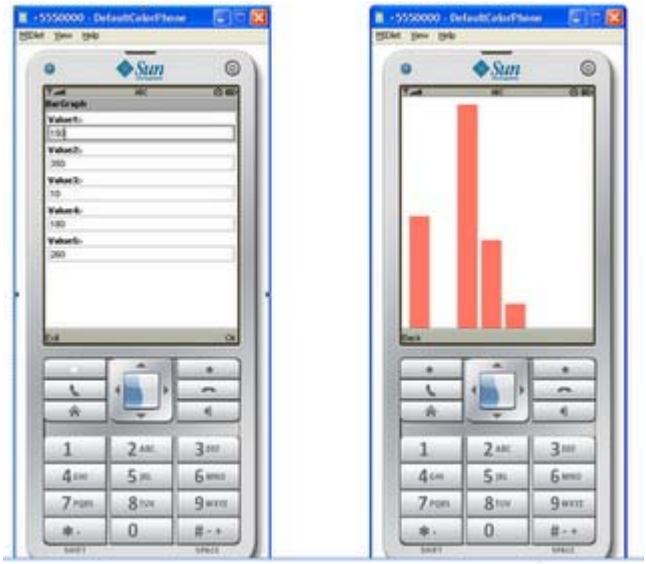
Create a MIDP application,which draws a bargraph to display.Data values can be given at int[]array?
Source Code:

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

public class BarGraph extends MIDlet implements CommandListener{
    public Form form;
    public Command exitCommand;
    public Command OkCommand;
    public Command backCommand;
    public Displayable d;
    public Display display;
    public TextField textfield1;
    public TextField textfield2;
    public TextField textfield3;
    public TextField textfield4;
    public TextField textfield5;
    public BarGraph ()
    {
```

```
    display=Display.getDisplay(this);
    form=new Form("BarGraph");
    textfield1=new TextField("Value1:-","",30,TextField.ANY);
    textfield2=new TextField("Value2:-","",30,TextField.ANY);
    textfield3=new TextField("Value3:-","",30,TextField.ANY);
    textfield4=new TextField("Value4:-","",30,TextField.ANY);
    textfield5=new TextField("Value5:-","",30,TextField.ANY);
    form.append(textfield1);
    form.append(textfield2);
    form.append(textfield3);
    form.append(textfield4);
    form.append(textfield5);
    OkCommand=new Command("Ok",Command.OK,1);
    exitCommand=new Command("Exit",Command.EXIT,1);
    backCommand=new Command("Back",Command.BACK,1);
    form.addCommand(OkCommand);
    form.addCommand(exitCommand);
    form.setCommandListener(this);
}
public void startApp() {
    display.setCurrent(form);
}
public void pauseApp() {
}
public void destroyApp(boolean unconditional) {
}
public void commandAction(Command command,Displayable displayable)
{
            if(displayable==form)
            {
              if(command==OkCommand)
               {
                 int[] data=new int[5];
                 data[0]=Integer.parseInt(textfield1.getString());
                 data[1]=Integer.parseInt(textfield2.getString());
                 data[2]=Integer.parseInt(textfield3.getString());
                 data[3]=Integer.parseInt(textfield4.getString());
                 data[4]=Integer.parseInt(textfield5.getString());
                 d=new BarCanvas(data);
                 d.addCommand(backCommand);
                 d.setCommandListener(this);
                 display.setCurrent(d);
               }
              else if(command==exitCommand)
                    notifyDestroyed();
               }
```

```
                    else if(displayable==d)
                   {
                    if(command==backCommand)
                     display.setCurrent(form);
                    }

                    }
}

class BarCanvas extends Canvas{
   int[] data;
   public int x;
   public int y;
   public int y1;
   public int h;
   public BarCanvas(int[] data)
   {
      this.data=data;
      x=10;
   }
public void paint(Graphics g)
   {
   g.setColor(255, 255, 255);
   g.fillRect(0, 0, this.getWidth(), this.getHeight());
    g.setColor(255, 125, 100);
    int i=0;
    y1=data[0];
    h=200;
    while(i<data.length)</data.length)
    {
       y=data[i];
       h=200+y1-y;
       g.fillRect(x, y,25 , h);
       x+=30;
       i++;
    }
     }
         }
```

Output:



4.3 Create a MIDP application, which draws a bar graph to the display. Data values can be given at int[] array. You can enter four data (integer) values to the input text field.

Create a MIDP Application,which draws a Pie Graph to the display.Data Values can be given at int[] array.You can enter four data(integer)values to the input text field

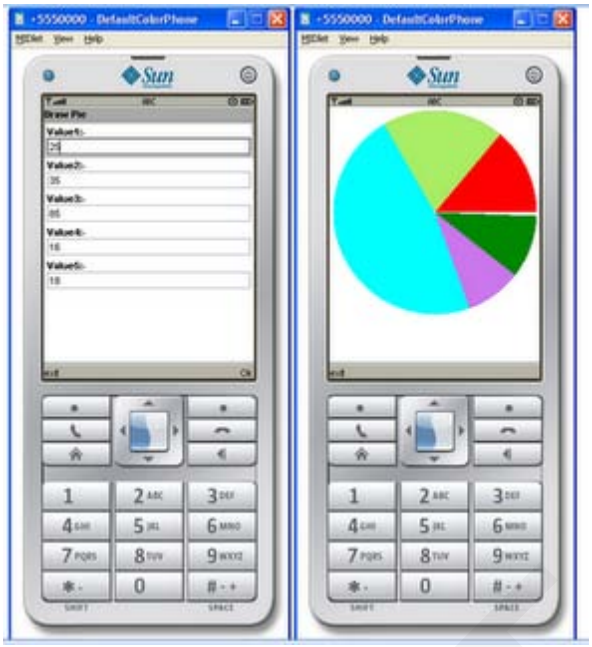
Create a MIDP Application,which draws a Pie Graph to the display.Data Values can be given at int[] array.You can enter four data(integer)values to the input text field.
Source Code:

```
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.*;
public class PieChart extends MIDlet implements CommandListener {
  public Form form;
  public Command exitCommand;
  public Command OkCommand;
  public Display display;
  public TextField textfield1;
  public TextField textfield2;
  public TextField textfield3;
  public TextField textfield4;
  public TextField textfield5;
public Displayable d;
  public void startApp() {
     display = Display.getDisplay(this);
      form=new Form("Draw Pie");
     textfield1=new TextField("Value1:-","",30,TextField.ANY);
```

```
    textfield2=new TextField("Value2:-","",30,TextField.ANY);
    textfield3=new TextField("Value3:-","",30,TextField.ANY);
    textfield4=new TextField("Value4:-","",30,TextField.ANY);
    textfield5=new TextField("Value5:-","",30,TextField.ANY);
   form.append(textfield1);
   form.append(textfield2);
   form.append(textfield3);
   form.append(textfield4);
   form.append(textfield5);
   exitCommand = new Command("exit", Command.EXIT, 1);
   OkCommand=new Command("Ok",Command.OK,1);
   form.addCommand(OkCommand);
   form.addCommand(exitCommand);
   form.setCommandListener(this);
   display.setCurrent(form);
 }
 public void pauseApp() {
 }
 public void destroyApp(boolean unconditional) {
 }
 public void commandAction(Command c, Displayable s) {
     if(s==form)
     {
     if(c==exitCommand)
   notifyDestroyed();
     else if(c==OkCommand)
     {
         int[] data = new int[5];

   data[0]=Integer.parseInt(textfield1.getString());
   data[1]=Integer.parseInt(textfield2.getString());
   data[2]=Integer.parseInt(textfield3.getString());
   data[3]=Integer.parseInt(textfield4.getString());
   data[4]=Integer.parseInt(textfield5.getString());
    d = new PieChartCanvas(data);
   d.addCommand(exitCommand);
   d.setCommandListener(this);
        display.setCurrent(d);
     }
     }
else if(s==d)
     {
    if(c==exitCommand)
       display.setCurrent(form);
}
 }
```

```
}

class PieChartCanvas extends Canvas {
  int[] data;

  int colors[] = { 0xFF0000, 0xA9E969, 0x00FFFF, 0xC675EC, 0x008800, 0x00C400 };

  public PieChartCanvas(int[] data) {
    this.data = data;
  }

  public void paint(Graphics g) {
    int width = this.getWidth();
    int height = this.getHeight();
    g.setColor(255, 255, 255);
    g.fillRect(0, 0, width, height);
    int sum = 0;
    for (int i = 0; i < data.length; i++) {
      sum += data[i];
    }
    int deltaAngle = 360 * 100 / sum / 100;
    int x = 4;
    int y = 4;
    int diameter;
    if (width > height)
      diameter = height - y * 2;
    else
      diameter = width - x * 2;
    int startAngle = 0;
    for (int i = 0; i < data.length; i++) {
      g.setColor(colors[i]);
      g.fillArc(x, y, diameter, diameter, startAngle, deltaAngle * data[i]);
      startAngle += deltaAngle * data[i];
    }
}
}
```

Output:



Week - 5 Developing Networked Applications using the Wireless Toolkit

Creating a Simple Client-Server Application

Create, compile and run a basic UDP-based client-server application.

Creating the Datagram Server project

1) Click on Wireless Toolkit 2.5.2 under the group: **All Programs→Sun Java**

(TM) Wireless Toolkit 2.5.2.

2) Click on '**New Project...**' button.

3) Enter project name as '**DatagramServer**'. Enter MIDlet name as '**DatagramServer**'. Note that the Midlet name is the same as the name of the class in the source code, which extends the MIDlet class, otherwise the application won't run.

4) Another window pops up where it is required to select a target platform. Select '**MIDP 1.0**' from the drop down list.

5) After clicking OK, the project is created; and the Wireless Toolkit tells that the name of the folder where source code files are created. The path of the source code folder is displayed in the debug output window.

Creating and Compiling the DatagramServer source files

The Wireless Toolkit does not come with an IDE by default so Use any IDE or a text editor like *Notepad*.

1) Create a new text file called **DatagramServer.java** in the source folder of the project. The exact path of this folder is displayed in the Wireless Toolkit window.

2) Paste contents **DatagramServer.java** from into the source file.

Running your Server application on the Phone simulator

1) After compiling the project successfully, click on the Run button in the Wireless Toolkit window.

2) A graphical window depicting a phone handset will appear with the name of your application highlighted on its screen as shown below.

3) To start the application, click on the right soft-key (marked with a dot) below the '**Launch'** command.

4) The phone simulator might ask if it is OK to run the network application. Select '**Yes**' by clicking on the appropriate soft-key. The server is now up and running.

5) Keep the server running during the creation, compilation and running of the Datagram Client application.

Creating the DatagramClient project

1) Use the same instance of the Wireless Toolkit that is used for creating and compiling the Datagram Server project.

2) Click on '**New Project...**' button.

3) A new window pops up. Enter project name as '**DatagramClient**'. Enter MIDlet name as '**DatagramClient**'. Note that the Midlet name is the same as the name of the class in the source code, which extends the MIDlet class.

4) Another window pops up where one has to select a target platform. Select '**MIDP 1.0**' from the drop down list.

5) After clicking OK, the project is created and the Wireless Toolkit tells where to place the source code files. The path of the source code folder is displayed in the debug output window as explained before.

Creating and Compiling the DatagramClient source files

1) Create a new text file called **DatagramClient.java** in the source folder of the

project.

2) Paste contents **DatagramClient.java** into the source file.

3) Then click on the Build button in the Wireless Toolkit window. If the compilation is OK, it will say Build Complete in the window's debug output window, otherwise it will show the errors. Note: In the source code, use the System.out.println() statement to output debug information to this window.

Running your Client application on the Phone simulator

1) After compiling the project successfully, click on the Run button in the Wireless Toolkit window.

2) A graphical window depicting a phone handset will appear with the name of the application highlighted on its screen.

3) To start the application, click on the right soft-key (marked with a dot) below the '**Launch**' command.

4) The phone simulator might ask if it is OK to run the network application. Select '**Yes**' by clicking on the appropriate soft-key. The client is now up and running.

5) When the client executes on the phone simulator, one should see a text box with the caption 'Message'. Enter any message and press the right soft-key (corresponding to Send). If the client-server application is working properly, the screen of the server phone will display the message sent by the client and the client screen will now display a message sent by the server in response. The response message from the server is the original client message in reverse.

6) Try various features of the phone simulator including the different look-and feel options.

Creating a Simple Client-Server Application

Developing Networked Applications using the Wireless Toolkit

Creating a Simple Client-Server Application

Create, compile and run a basic UDP-based client-server application.

Creating the Datagram Server project

1) Click on Wireless Toolkit 2.5.2 under the group: **All Programs→Sun Java**

(TM) Wireless Toolkit 2.5.2.

2) Click on '**New Project...**' button.

3) Enter project name as '**DatagramServer**'. Enter MIDlet name as '**DatagramServer**'. Note that the Midlet name is the same as the name of the class in the source code, which extends the MIDlet class, otherwise the application won't run.

4) Another window pops up where it is required to select a target platform. Select '**MIDP 1.0**' from the drop down list.

5) After clicking OK, the project is created; and the Wireless Toolkit tells that the name of the folder where source code files are created. The path of the source code folder is displayed in the debug output window.

Creating and Compiling the DatagramServer source files

The Wireless Toolkit does not come with an IDE by default so Use any IDE or a text editor like *Notepad*.

1) Create a new text file called **DatagramServer.java** in the source folder of the project. The exact path of this folder is displayed in the Wireless Toolkit window.

2) Paste contents **DatagramServer.java** from into the source file.

Running your Server application on the Phone simulator

1) After compiling the project successfully, click on the Run button in the Wireless Toolkit window.

2) A graphical window depicting a phone handset will appear with the name of your application highlighted on its screen as shown below.

3) To start the application, click on the right soft-key (marked with a dot) below the '**Launch'** command.

4) The phone simulator might ask if it is OK to run the network application. Select '**Yes**' by clicking on the appropriate soft-key. The server is now up and running.

5) Keep the server running during the creation, compilation and running of the Datagram Client application.

Source code:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

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

```
import javax.microedition.io.*;


/**
 * @author ADMIN
 */
public class DatagramServer extends MIDlet implements CommandListener{
    public Form form1;
    public Form form2;
    public Command startCommand;
    public Command refreshCommand;
    public Command exitCommand;
    public Display display;
    public StringItem st;
    public DatagramServer()
    {
        display=Display.getDisplay(this);
        startCommand=new Command("Start",Command.OK,1);
        refreshCommand=new Command("Refresh",Command.OK,1);
        exitCommand=new Command("Exit",Command.EXIT,1);
        st=new StringItem(" "," ");
        form1 =new Form("DataGramserver");
        form2=new Form("Ready to receive Messages");
        form1.addCommand(startCommand);
        form1.setCommandListener(this);
        form2.addCommand(refreshCommand);
```

```
    form2.addCommand(exitCommand);
    form2.setCommandListener(this);
  }
  public void startApp() {
    display.setCurrent(form1);
  }

  public void pauseApp() {
  }

  public void destroyApp(boolean unconditional) {
  }

  public void commandAction(Command cmd,Displayable displayable)
  {
    if(displayable==form1)
    {
      if(cmd==startCommand)
      {
        try {
DatagramConnection dgc = (DatagramConnection)
  Connector.open("datagram://:9001");
try {
  int size = 100;
  Datagram datagram = dgc.newDatagram(size);
```

```
    dgc.receive(datagram);


    form2.append(datagram.getData().toString());
   } finally {
      dgc.close();
   }
 } catch (Exception x){
   x.printStackTrace();
 }
```



```
            display.setCurrent(form2);
          }
        }
  else if(displayable==form2)
        {
          if(cmd==exitCommand)
          {
            notifyDestroyed();
          }
  else if(cmd==refreshCommand)
          {
    st.setText(" ");
  }
```

```
    }
  }
}
```

Creating the DatagramClient project

1) Use the same instance of the Wireless Toolkit that is used for creating and compiling the Datagram Server project.

2) Click on '**New Project...**' button.

3) A new window pops up. Enter project name as '**DatagramClient**'. Enter MIDlet name as '**DatagramClient**'. Note that the Midlet name is the same as the name of the class in the source code, which extends the MIDlet class.

4) Another window pops up where one has to select a target platform. Select '**MIDP 1.0**' from the drop down list.

5) After clicking OK, the project is created and the Wireless Toolkit tells where to place the source code files. The path of the source code folder is displayed in the debug output window as explained before.

Creating and Compiling the DatagramClient source files

1) Create a new text file called **DatagramClient.java** in the source folder of the

project.

2) Paste contents **DatagramClient.java** into the source file.

3) Then click on the Build button in the Wireless Toolkit window. If the compilation is OK, it will say Build Complete in the window's debug output window, otherwise it will show the errors. Note: In the source code, use the System.out.println() statement to output debug information to this window.

Running your Client application on the Phone simulator

1) After compiling the project successfully, click on the Run button in the Wireless Toolkit window.

2) A graphical window depicting a phone handset will appear with the name of the application highlighted on its screen.

3) To start the application, click on the right soft-key (marked with a dot) below the '**Launch**' command.

4) The phone simulator might ask if it is OK to run the network application. Select '**Yes**' by clicking on the appropriate soft-key. The client is now up and running.

5) When the client executes on the phone simulator, one should see a text box with the caption 'Message'. Enter any message and press the right soft-key (corresponding to Send). If the client-server application is working properly, the screen of the server phone will display the message sent by the client and the client screen will now display a message sent by the server in response. The response message from the server is the original client message in reverse.

6) Try various features of the phone simulator including the different look-and feel options.

Source code

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
/**
 * @author ADMIN
 */
public class DatagramClient extends MIDlet implements CommandListener{
    public Form form1;
    public Display display;
    public TextField textfield;
```

```
public Command sendCommand;
public DatagramClient()
{
   display=Display.getDisplay(this);
   form1=new Form("Datagram Client");
   sendCommand=new Command("send",Command.OK,1);
   textfield=new TextField("Enter Text",null,30,TextField.ANY);
   form1.append(textfield);
   form1.addCommand(sendCommand);
   form1.setCommandListener(this);
}
public void startApp() {
   display.setCurrent(form1);
}

public void pauseApp() {
}

public void destroyApp(boolean unconditional) {
}
public void commandAction(Command cmd,Displayable d)
{
   if(cmd==sendCommand)
   {
      try         {
```

```
  DatagramConnection dgc = (DatagramConnection)
    Connector.open("datagram://localhost:9001");
  try {
while(true)
{
    byte[] payload = textfield.getString().getBytes();
    Datagram datagram = dgc.newDatagram(payload, payload.length);
    dgc.send(datagram);
}
  } finally {
      dgc.close();
  }
} catch (Exception x) {
  x.printStackTrace();
}
```

```
      }
    }
}
```

Week - 6 Authentication with a Web Server

6.1 Login to HTTP Server from a J2ME Program

This J2ME sample program shows how to display a simple LOGIN SCREEN on the J2ME phone and how to authenticate to a HTTP server.

Many J2ME applications for security reasons require the authentication of the user. This free J2ME sample program, shows how a J2ME application can do authentication to the backend

server.
Note: Use Apache Tomcat Server as Web Server and Mysql as Database Server.

Login to HTTP Server from a J2ME Program

Login to HTTP Server from a J2ME Program

This J2ME sample program shows how to display a simple LOGIN SCREEN on the J2ME phone and how to authenticate to a HTTP server.

Many J2ME applications for security reasons require the authentication of the user. This free J2ME sample program, shows how a J2ME application can do authentication to the backend server.

Note: Use Apache Tomcat Server as Web Server and Mysql as Database Server.

Source code:

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import java.io.*;
/**
 * @author MCA
 */
public class login extends MIDlet implements CommandListener {
    public Form form1;
    public Command okCommand;
    public Display display;
    public HttpConnection ht=null;
    public InputStream ist=null;
    public StringItem st;
    public TextField t1;
    public TextField t2;
    public Alert alert;
    public Form form2;
    public login()
    {
        display=Display.getDisplay(this);
        st=new StringItem(" "," Welcome");
        alert =new Alert(" ","Wrong UserName or Password",null,AlertType.INFO);

        t1=new TextField("UserName"," ",30,TextField.ANY);
        t2=new TextField("Password"," ",30,TextField.PASSWORD);
      form1=new Form("Login Here");
```

```
 form2=new Form("Welcome");
 okCommand=new Command("Login",Command.OK,1);
 form1.addCommand(okCommand);
 form1.setCommandListener(this);

 form1.append(t1);
 form1.append(t2);
 form2.append(st);
}

public void startApp() {
   display.setCurrent(form1);
}

public void pauseApp() {
}

public void destroyApp(boolean unconditional) {
   notifyDestroyed();
}
public void commandAction(Command cmd,Displayable d)
{
   if(cmd==okCommand)
   {
      try
      {

 //        String  url="http://192.168.5.19:8080/WebApplication7/index.jsp?t1=101&t2=aaa";
          String
url="http://192.168.5.19:8080/WebApplication7/index.jsp?t1="+t1.getString().trim()+"&t2="+t2
.getString().trim();

//ht=(HttpConnection)Connector.open("http://192.168.5.19:8080/WebApplication7/index.jsp");
          ht=(HttpConnection)Connector.open(url);
          ist=ht.openInputStream();
          byte[] b=new byte[900];
          ist.read(b);
          String s=new String(b);
          s=s.trim();
          if(s.equals("ok"))
          display.setCurrent(form2);
             else
           {

             alert.setTimeout(Alert.FOREVER);
          display.setCurrent(alert);
```

```
                }


            }
            catch(Exception ex)
            {
form1.append(ex.toString());
            }


        }
    }
}
```