MOBILE APPLICATION DEVELOPMENT

LABORATORY MANUAL

B.TECH

(IV YEAR – I SEM)

(2019-20)

Department of Information Technology

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

Recognized under 2(f) and 12 (B) of UGC ACT 1956

Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015 Certified)

Maisammaguda, Dhulapally (Post Via. Kompally), Secunderabad – 500100, Telangana State, India

DEPARTMENT OF INFORMATION TECHNOLOGY

VISION

- To improve the quality of technical education that provides efficient software engineers with an attitude to adapt challenging IT needs of local, national and international arena, through teaching and interaction with alumni and industry.

MISSION

- Department intends to meet the contemporary challenges in the field of IT and is playing a vital role in shaping the education of the 21st century by providing unique educational and research opportunities.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1 – ANALYTICAL SKILLS

To facilitate the graduates with the ability to visualize, gather information, articulate, analyze, solve complex problems, and make decisions. These are essential to address the challenges of complex and computation intensive problems increasing their productivity.

PEO2 – TECHNICAL SKILLS

To facilitate the graduates with the technical skills that prepare them for immediate employment and pursue certification providing a deeper understanding of the technology in advanced areas of computer science and related fields, thus encouraging to pursue higher education and research based on their interest.

PEO3 – SOFT SKILLS

To facilitate the graduates with the soft skills that include fulfilling the mission, setting goals, showing self-confidence by communicating effectively, having a positive attitude, get involved in team-work, being a leader, managing their career and their life.

PEO4 – PROFESSIONAL ETHICS

To facilitate the graduates with the knowledge of professional and ethical responsibilities by paying attention to grooming, being conservative with style, following dress codes, safety codes, and adapting themselves to technological advancements.

PROGRAM SPECIFIC OUTCOMES (PSOs)

After the completion of the course, B. Tech Information Technology, the graduates will have the following Program Specific Outcomes:

1. **Fundamentals and critical knowledge of the Computer System:-** Able to Understand the working principles of the computer System and its components , Apply the knowledge to build, asses, and analyze the software and hardware aspects of it .

2. **The comprehensive and Applicative knowledge of Software Development:** Comprehensive skills of Programming Languages, Software process models, methodologies, and able to plan, develop, test, analyze, and manage the software and hardware intensive systems in heterogeneous platforms individually or working in teams.

3. **Applications of Computing Domain & Research:** Able to use the professional, managerial, interdisciplinary skill set, and domain specific tools in development processes, identify the research gaps, and provide innovative solutions to them.

PROGRAM OUTCOMES (POs)

Engineering Graduates should possess the following:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design / development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multi disciplinary environments.
12. **Life- long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

Maisammaguda, Dhulapally Post, Via Hakimpet, Secunderabad – 500100

DEPARTMENT OF INFORMATION TECHNOLOGY

GENERAL LABORATORY INSTRUCTIONS

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time), those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis / program / experiment details.
3. Student should enter into the laboratory with:

a. Laboratory observation notes with all the details (Problem statement, Aim, Algorithm, Procedure, Program, Expected Output, etc.,) filled in for the lab session.

b. Laboratory Record updated up to the last session experiments and other utensils (if any) needed in the lab.

c. Proper Dress code and Identity card.

4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.
5. Execute your task in the laboratory, and record the results / output in the lab observation note book, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must maintain the discipline and decency in the laboratory.
7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
9. Students must take the permission of the faculty in case of any urgency to go out; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the lab after completing the task (experiment) in all aspects. He/she must ensure the system / seat is kept properly.

HEAD OF THE DEPARTMENT **PRINCIPAL**

INDEX

1. Installation of Java Wireless Toolkit (J2ME) DATE:

1) Along with Java Runtime Environment (JRE) installed, install the latest Java Development Kit (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:

Start -> All Programs -> Sun Java Tool Kit -> Wireless Tool Kit
Click New Project – Enter Project Name -> Enter Class Name -> Click on Create Project. Choose appropriate API Selection and Configurations.

Place Java Source file in WTK2.1 / WTK2.2\apps\projectname\src Build the Project.

Run the Project.

2. Hello World DATE:

```
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();
  }
}
```

3. Menu Creation DATE:

Lets create a program with following kind of menu.

* **cut**
* **copy**
* **past**
* **delete**
* **select all**
* **unselect all**

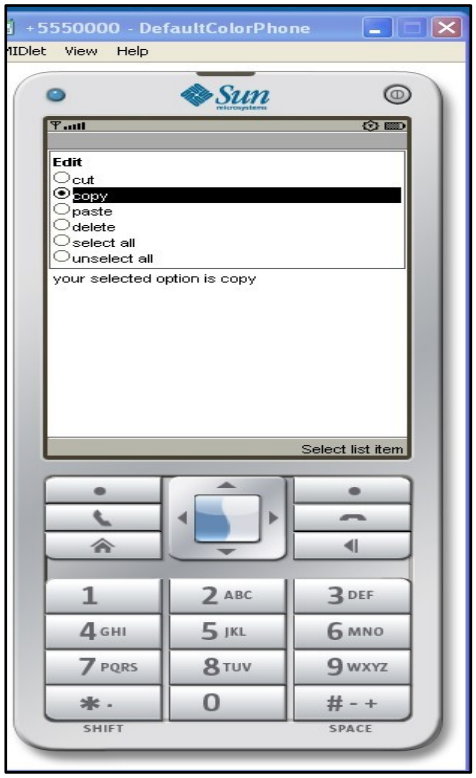
Program:

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);
}
}
}
```

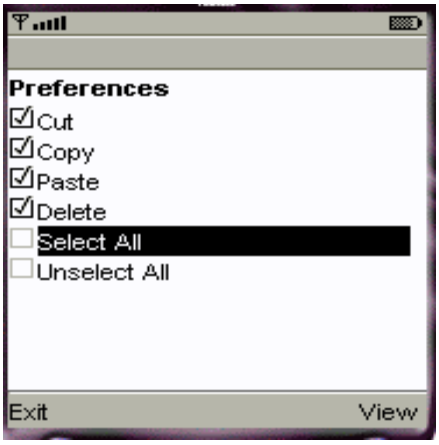
Output:



4. Menu Events Handling **DATE:**

Lets create a menu with 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**



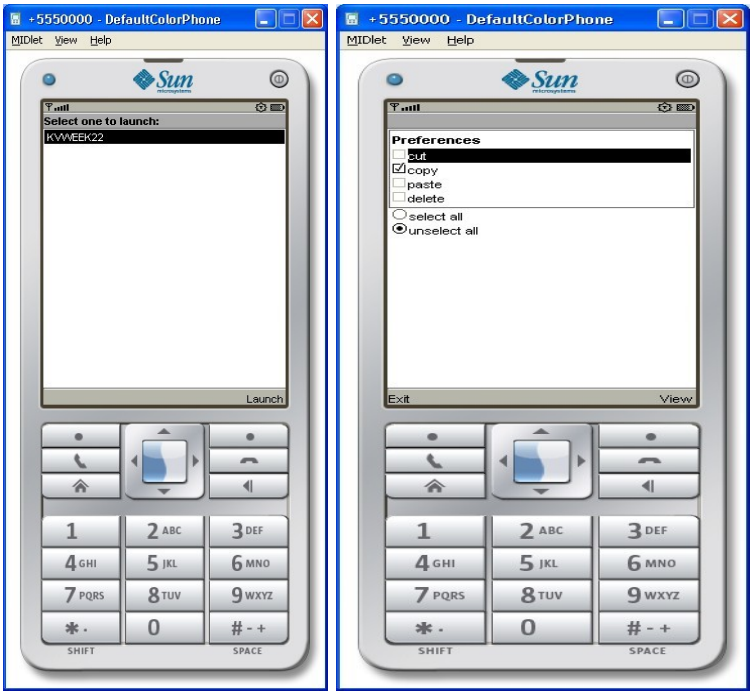
Program:

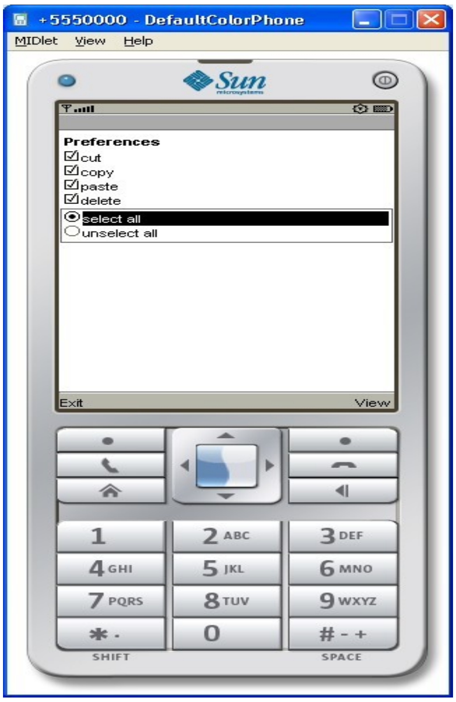
```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class MenuEvents extends MIDlet implements CommandListener,ItemStateListener { public
ChoiceGroup ch;
public ChoiceGroup ch1;
public Form form; public
Form form1; public
Display display; public
Command View; public
Command 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);
```

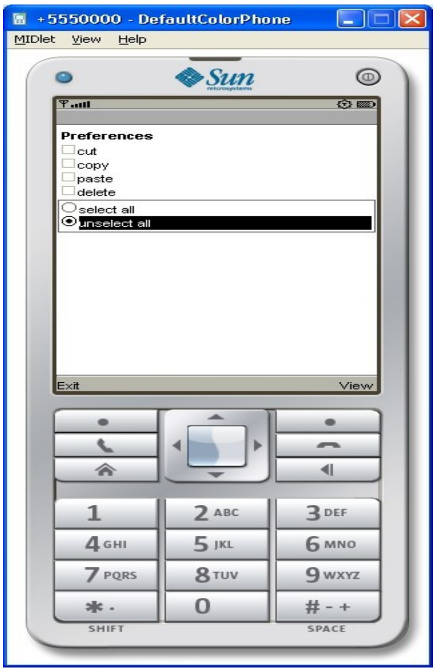
```
ch1.append("select all",null); ch1.append("unselect all",null); ch1.setSelectedIndex(1, true);
  form.append(ch1);
View=new Command("View",Command.OK,1); Exit
=new Command("Exit",Command.EXIT,1); Back=new
Command("Back",Command.BACK,1);
form.addCommand(View); form.addCommand(Exit);
form1.addCommand(Back);
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)
```

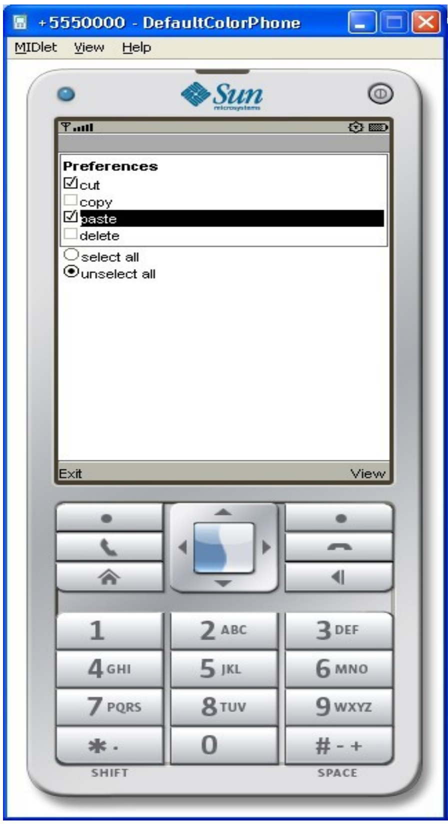
```
{
if(command==Back)
{
display.setCurrent(form);
options.setText("");
}
}
}
public void itemStateChanged(Item item)
{
if(item==ch1)
{
int i=0;
int size=ch.size(); while(i<size)<
p=""> </size)<>
{
if(ch1.getSelectedIndex()==0)
ch.setSelectedIndex(i, true); else
ch.setSelectedIndex(i, false);
i++;
}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

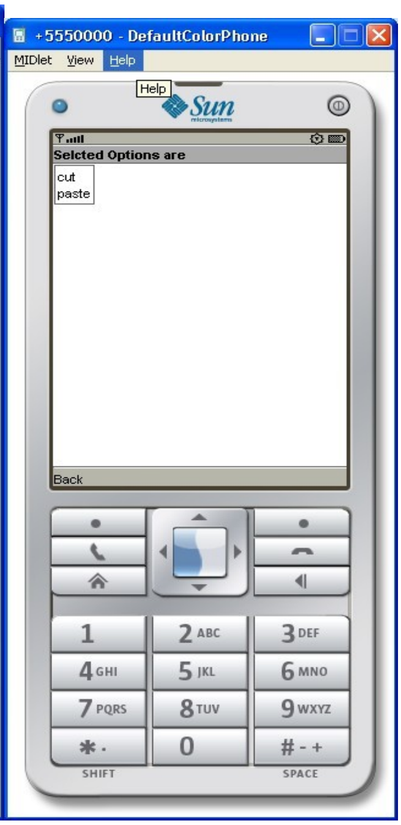
Output:

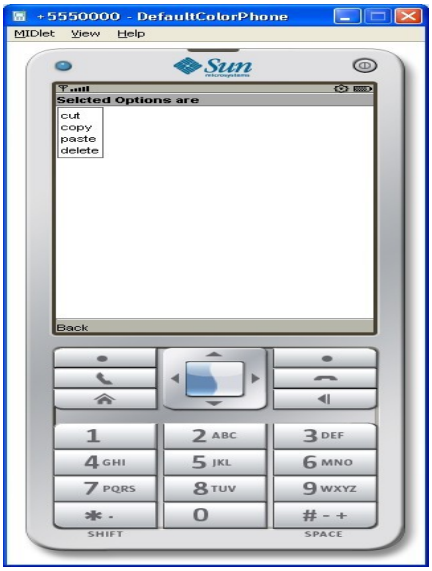


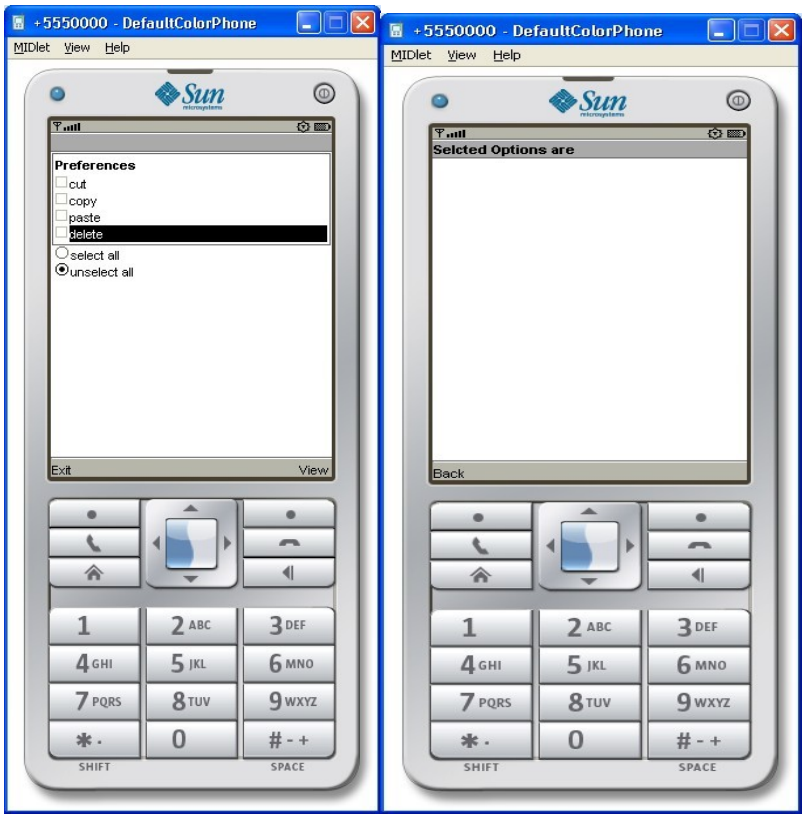

+5550000 - DefaultColorPhone
MIDlet View Help
Preferences
cut
copy
paste
delete
select all
unselect all
Exit
View


+5550000 - DefaultColorPhone
MIDlet View Help
Preferences
cut
copy
paste
delete
select all
unselect all
Exit
View


+5550000 - DefaultColorPhone
MIDlet View Help
Preferences
cut
copy
paste
delete
select all
unselect all
Exit
View


+5550000 - DefaultColorPhone
MIDlet View Help
Help
Selcted Options are
cut
paste
Back


+5550000 - DefaultColorPhone
MIDlet View Help
Selcted Options are
cut
copy
paste
delete
Back
1 2 ABC 3 DEF
4 GHI 5 JKL 6 MNO
7 PQRS 8 TUV 9 WXYZ
* . 0 # - +
SHIFT SPACE


+5550000 - DefaultColorPhone
MIDlet View Help
Preferences
cut
copy
paste
delete
select all
unselect all
Exit View
+5550000 - DefaultColorPhone
MIDlet View Help
Selcted Options are
Back
1 2 ABC 3 DEF
4 GHI 5 JKL 6 MNO
7 PQRS 8 TUV 9 WXYZ
* . 0 # - +
SHIFT SPACE

5. Slide Show DATE:

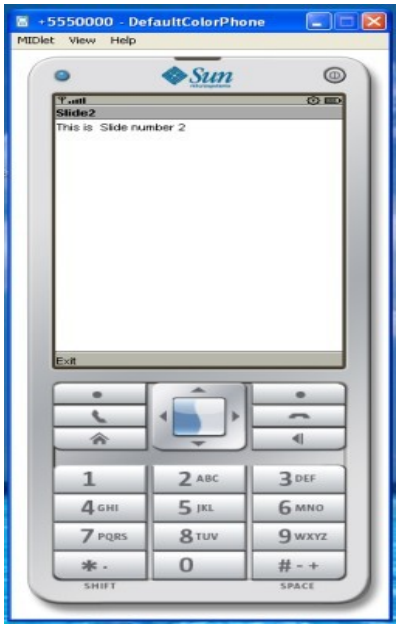
Let's create a slide show which has three slides, which includes only text. Program should change to the new slide after 5 seconds. After the third slide, program returns to the first slide.


```
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:-



6. Image Slide Show DATE:

Let's create a slide show which has three slides, which includes pictures at PNG format. Program should change to the new slide other 5 seconds.


```
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 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);
```

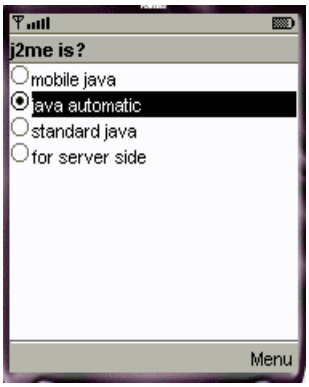
```
slide2=new Form("Slide2");
slide2.append(imageitem2);
slide2.addCommand(Exit); slide3=new
Form("Slide3");
slide3.append(imageitem3);
slide3.addCommand(Exit);
slide1.setCommandListener(this);
slide2.setCommandListener(this);
slide3.setCommandListener(this);
}

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:-

7. (A) Quiz DATE:

Lets create a MIDP application, that shows 5-10 quiz questions to the user. All questions have 4 possible options and one right option exactly. Application counts and shows the count of right answers to the user.



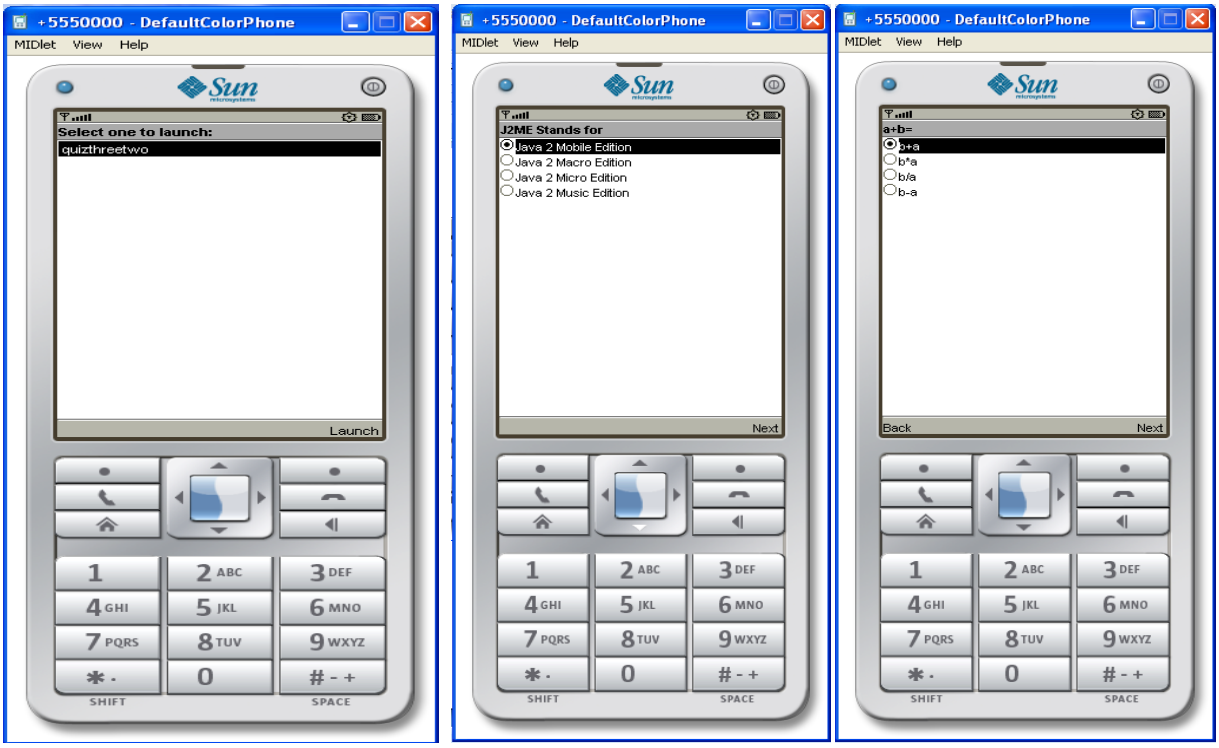

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.rms.*;
import java.io.*;
public class Quiz extends MIDlet implements CommandListener {
public Form form1;
public Form form2;
public Form form3;
public Form form4;
public Form form5;
public Form form6;
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()
```

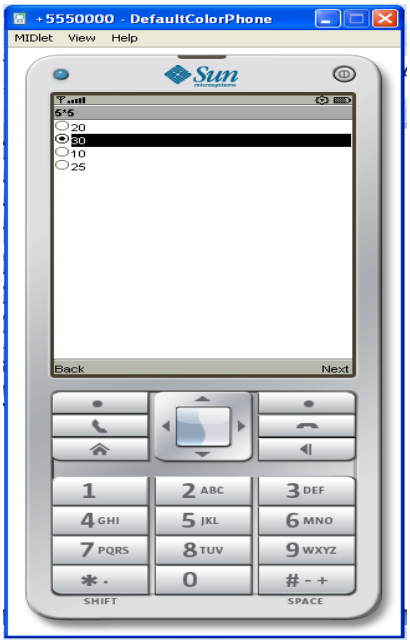
```
{
count=0;
display=Display.getDisplay(this);
nextCommand=new Command("Next",Command.OK,1);
backCommand=new Command("Back",Command.BACK,1);
st1=new StringItem("","");
textfield=new TextField("EnterName","",20,TextField.ANY);

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);
```

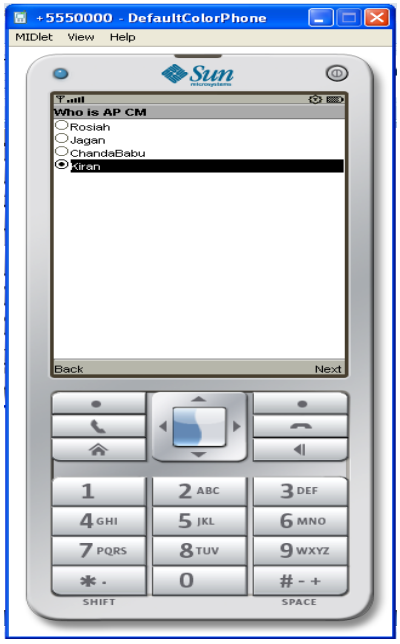
```
ch5.append("11", null);
ch5.append("23", null);

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++;
```

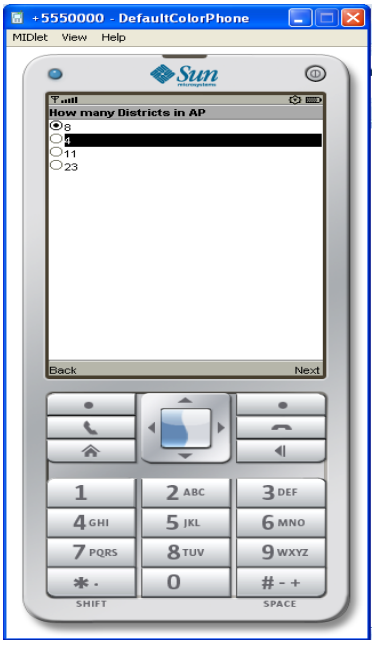
```
if(ch3.getSelectedIndex()==3)
count++;

if(ch4.getSelectedIndex()==3)
count++;

if(ch5.getSelectedIndex()==3)
count++;

st.setText(String.valueOf(count));

form6.append(st);
form6.append(textfield);
display.setCurrent(form6);

}
}
}
}
```

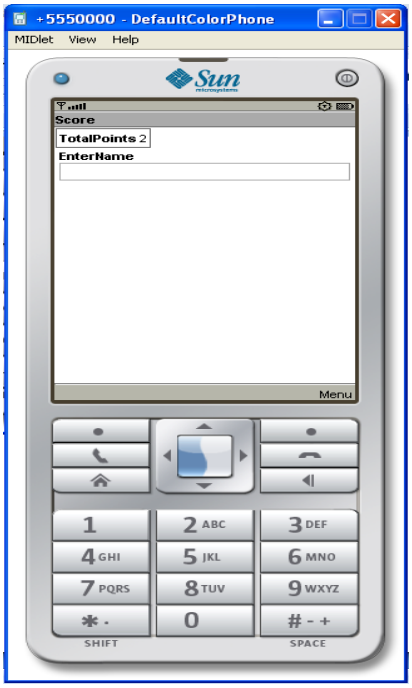
Output:

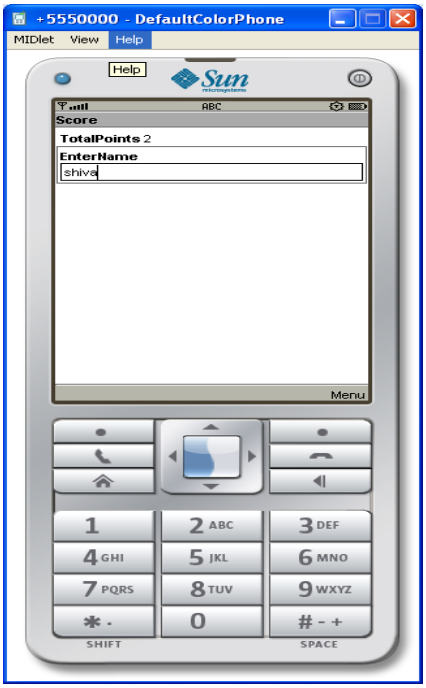


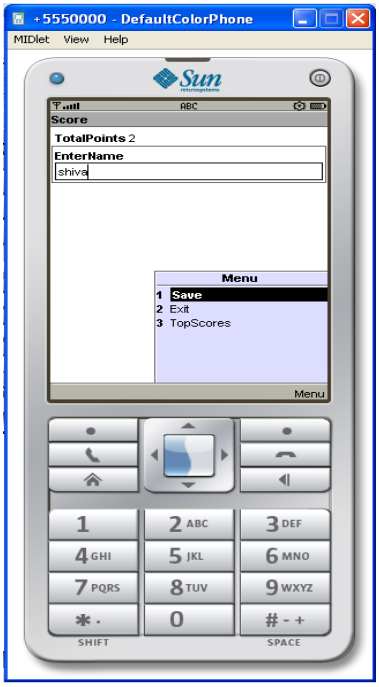

+5550000 - DefaultColorPhone
MIDlet View Help
5*5
20
30
10
25
Back
Next

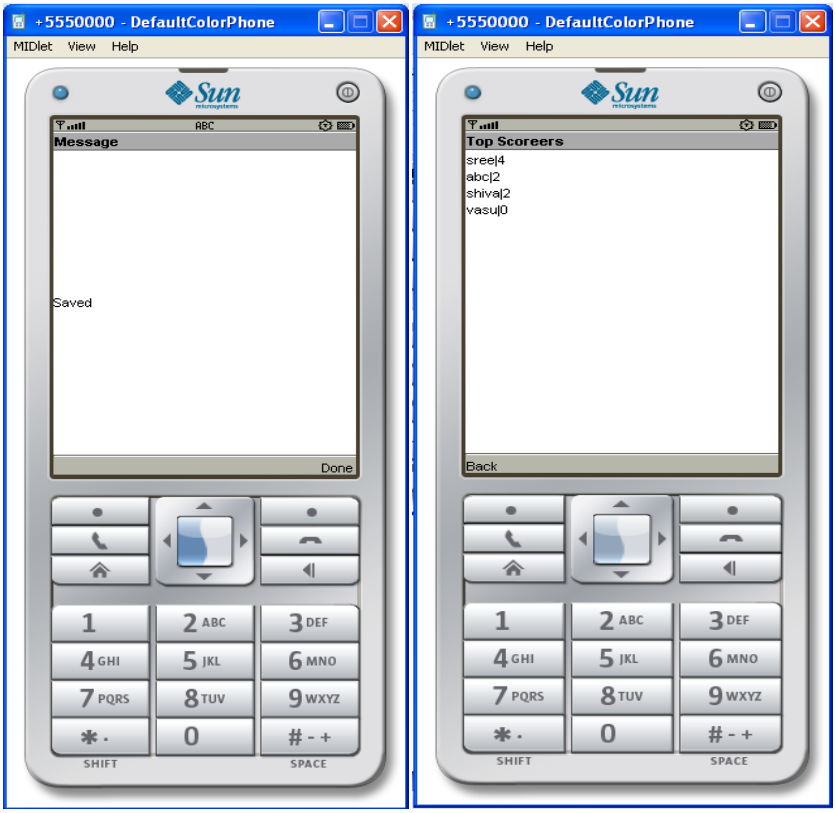

+5550000 - DefaultColorPhone
MIDlet View Help
Who is AP CM
Rosiah
Jagan
ChandaBabu
Kiran
Back
Next


+5550000 - DefaultColorPhone
MIDlet View Help
How many Districts in AP
8
11
23
Back
Next


+5550000 - DefaultColorPhone
MIDlet View Help
Score
TotalPoints 2
EnterName
Menu

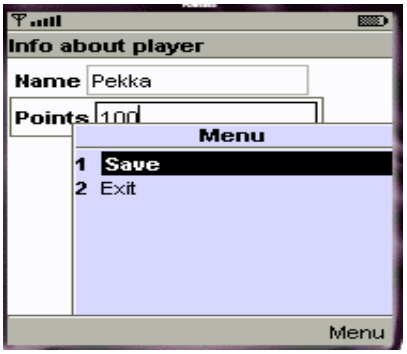

+5550000 - DefaultColorPhone
MIDlet View Help
Help
Score
TotalPoints 2
EnterName
shiva
Menu


+5550000 - DefaultColorPhone
MIDlet View Help
Score
TotalPoints 2
EnterName
shiva
Menu
1 Save
2 Exit
3 TopScores
Menu


+5550000 - DefaultColorPhone
MIDlet View Help
Message
Saved
Done
+5550000 - DefaultColorPhone
MIDlet View Help
Top Scoreers
sree|4
abc|2
shiva|2
vasu|0
Back

7. (B) RMS DATE:

Lets Create a MIDP application, where the user can enter player name and points. The program saves the information to the record using RMS at MIDP device. Program should also print out the top 10 player list to the end user. You can use this class in your game if you made own class for saving and reading record sets.




```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.rms.*;
import java.io.*;
public class QuizMidlet 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 Compare comp;
public StringItem st1;
public QuizMidlet()
{
count=0;
```

```
display=Display.getDisplay(this);
nextCommand=new Command("Next",Command.OK,1);
backCommand=new Command("Back",Command.BACK,1);
OkCommand=new Command("Save",Command.SCREEN,1);
ExitCommand=new Command("Exit",Command.SCREEN,1);
sCommand=new Command("TopScores",Command.SCREEN,1);
st=new StringItem("TotalPoints","0");
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");
form7=new Form("Top Scoreers");
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(OkCommand);
form6.addCommand(ExitCommand);
form6.addCommand(sCommand);
form6.setCommandListener(this);

form7.addCommand(backCommand);
form7.setCommandListener(this);
try
{
recordstore=RecordStore.openRecordStore("Quiz", true);
}
catch(Exception ex)
{
}
}
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)
```

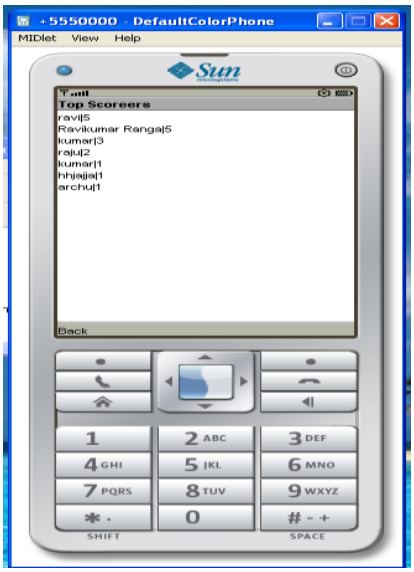
```
display.setCurrent(form5);
else if(cmd==backCommand)
display.setCurrent(form3);
}
else if(displayable==form5)
{
if(cmd==nextCommand)
{
if(ch1.getSelectedIndex()==2)
count++;


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);
}

else if(cmd==backCommand)
display.setCurrent(form4);
}
else if(displayable==form6)
{
if(cmd==OkCommand)
{

try
{
String Pname=textfield.getString();
int Points=Integer.parseInt(st.getText());
byte[] bytes;
ByteArrayOutputStream ostream=new ByteArrayOutputStream();
DataOutputStream dstream =new DataOutputStream(ostream);
dstream.writeUTF(Pname);
dstream.writeInt(Points);
dstream.flush();
bytes=ostream.toByteArray();
recordstore.addRecord(bytes, 0,bytes.length );
ostream.reset();
ostream.close();
dstream.close();
alert =new Alert("Message","Saved",null,AlertType.INFO);
alert.setTimeout(Alert.FOREVER);
display.setCurrent(alert);
}
catch(Exception ex)
{
```

```
alert =new Alert("Message",ex.toString(),null,AlertType.INFO);
alert.setTimeout(Alert.FOREVER);
display.setCurrent(alert);
}
}
else if(cmd==ExitCommand)
{
try
{
recordstore.closeRecordStore();
notifyDestroyed();
}
catch(Exception ex)
{
}
}
else if(cmd==sCommand)
{
try
{
byte[] bytes=new byte[300];
ByteArrayInputStream bstream=new ByteArrayInputStream(bytes);

DataInputStream dstream=new DataInputStream(bstream);
StringBuffer sb=new StringBuffer();
comp=new Compare();
re=recordstore.enumerateRecords(null, comp,false);
st1.setText("");

while(re.hasNextElement())
{
recordstore.getRecord(re.nextRecordId(), bytes, 0);
sb.append(dstream.readUTF()+"|"+dstream.readInt());
sb.append("\n");
dstream.reset();
}
bstream.close();
dstream.close();
st1.setText(sb.toString());
form7.append(st1);
}

catch(Exception ex)
{
alert =new Alert("Msg",ex.toString(),null,AlertType.INFO);
alert.setTimeout(Alert.FOREVER);
display.setCurrent(alert);
}
display.setCurrent(form7);
}
}
else if(displayable==form7)
{
if(cmd==backCommand)
display.setCurrent(form6);
} }}
```

```
class Compare implements RecordComparator
{
public byte[] bytedata=new byte[300];
public ByteArrayInputStream bstream=null;
public DataInputStream dstream=null;
public int compare(byte[] r1,byte[] r2)
{
int r1int,r2int;
int or=0;
try
{
int maxlen=Math.max(r1.length, r2.length);

if(maxlen>bytedata.length)
{
bytedata=new byte[maxlen];
}
bstream=new ByteArrayInputStream(r1);
dstream=new DataInputStream(bstream);
dstream.readUTF();
r1int=dstream.readInt();
bstream=new ByteArrayInputStream(r2);
dstream=new DataInputStream(bstream);
dstream.readUTF();
r2int=dstream.readInt();
if(r1int==r2int)
{
or= RecordComparator.EQUIVALENT;
}
else if(r1int>r2int)
{
or= RecordComparator.PRECEDES;
}
else if(r1int<r2int)
{
or= RecordComparator.FOLLOWS;
}
return or;
}
catch(Exception ex)
{
return RecordComparator.EQUIVALENT;
}
}
public void compareClose()
{
try
{
if(bstream!=null)
{
bstream.close();
}
if(dstream!=null)
{
dstream.close();
}
```

```
}
catch(Exception ex)
{
} }
}
```

Output:-



8. Input Checking DATE:

Lets 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)

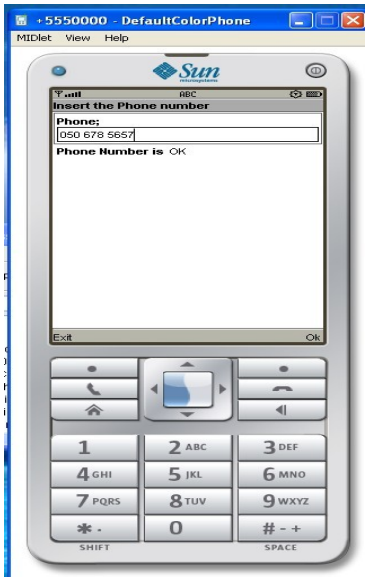
Program:

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

public class InputChecking extends MIDlet implements CommandListener {

    public Form form1;
    public TextField textfield1; public
    Command exitCommand; public
    Command okCommand; public
    StringItem st;
    public Display display;

    public InputChecking()
    {
        display=Display.getDisplay(this);
        form1=new Form("Insert the Phone number"); exitCommand=new
        Command("Exit",Command.EXIT,1); okCommand=new
        Command("Ok",Command.OK,1); st=new StringItem("Phone
        Number is ",""); textfield1=new
        TextField("Phone;","",30,TextField.ANY); form1.append(textfield1);
        form1.addCommand(okCommand);
        form1.addCommand(exitCommand);
        form1.setCommandListener(this);
    }
    public void startApp() {
        display.setCurrent(form1);
    }
    public void pauseApp() {
    }
```

```
public void destroyApp(boolean unconditional) {
}
public void commandAction(Command cmd,Displayable displayable)
{
    if(cmd==exitCommand)
        notifyDestroyed();
    else if(cmd==okCommand)
    {


        String s=textfield1.getString();
        s=s.replace(' ', '.');
        int len=s.length(); int
        i=0;
        int c=0;
         String s1="";
        while(i<len)                {

            if(s.charAt(i)=='.')
            {
                if(c==0)
                {
                if(s1.equals("040") || s1.equals("041") || s1.equals("050") || s1.equals("0400") || s1.equals("044"))
                {
                  c++;
                  s1="";
                }}
                  if(c==1)
                {

                   if(s1.length()-1==3)
                   {
                       c++;
                       s1="";
                    }}
            }
                   s1=s1+s.charAt(i);
                          i++;
        }

        if(s1.length()-1==3 || s1.length()-1==4 || s1.length()-1==5) c++;

        if(c==3)
            st.setText("OK");

            else
        {

            st.setText("wrong\n Phone Number Format is xxx xxx xxxx\nArea code must be
040|050|041|0400|044");
        }
```

```
            form1.append(st);

               }
            }
        }
```

Output:



9. Bar Graph DATE:

Lets create a MIDP application, which draws a bar graph to the 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() {
}
```

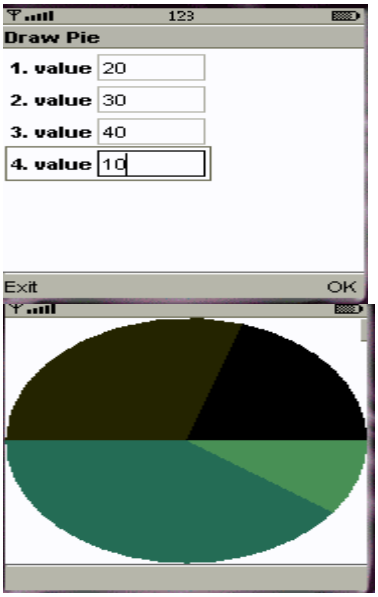
```
public void destroyApp(boolean unconditional) {
}

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:

10. Pie Chart DATE:

Lets 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.



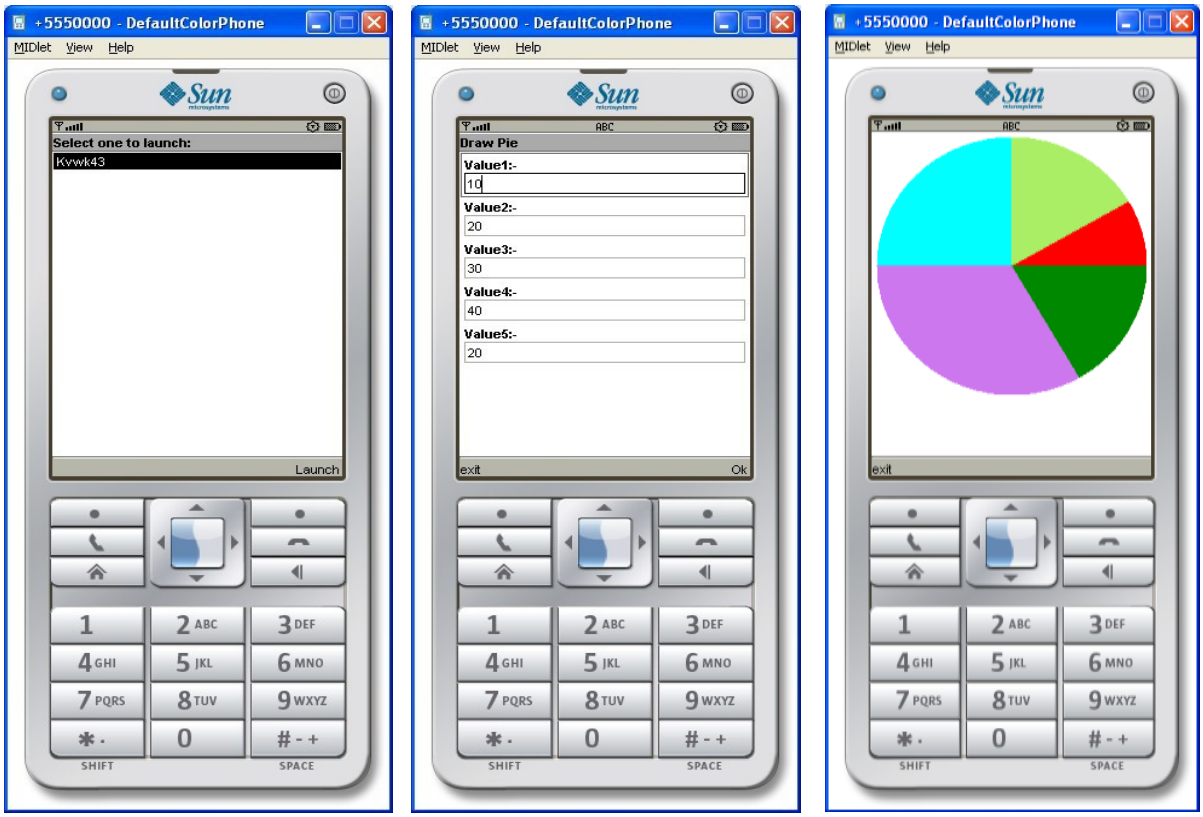
Program:

```
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:



11. Client Server App using Datagram DATE

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 Midlet name is 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.

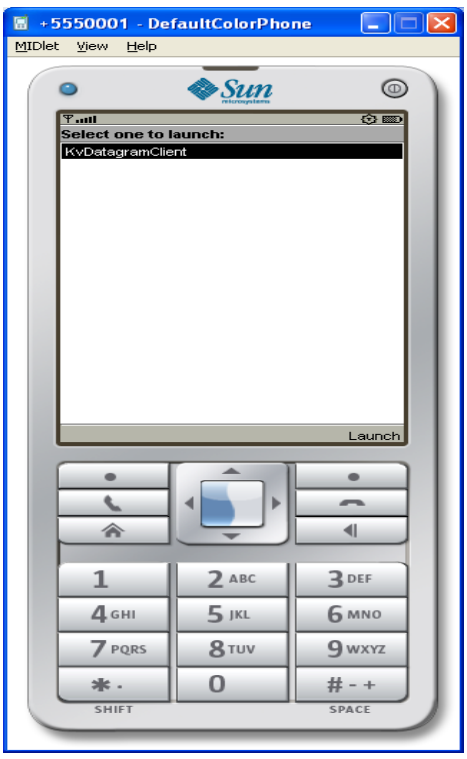
5) When the client executes on the phone simulator, one should see a text box with the caption 'Message'. Enter any message and press the right soft-key (corresponding to Send). If the client-server application is working properly, the screen of the server phone will display the message sent by the client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

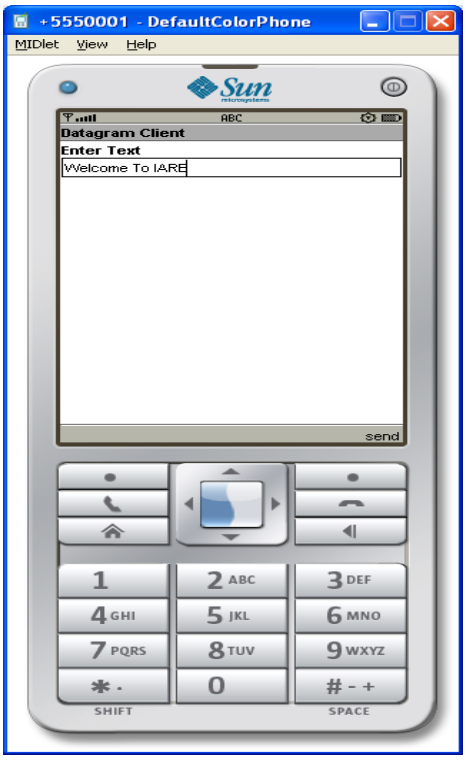
6) Try various features of the phone simulator including the different look-and feel options.

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
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();
}}}}
```

Output:

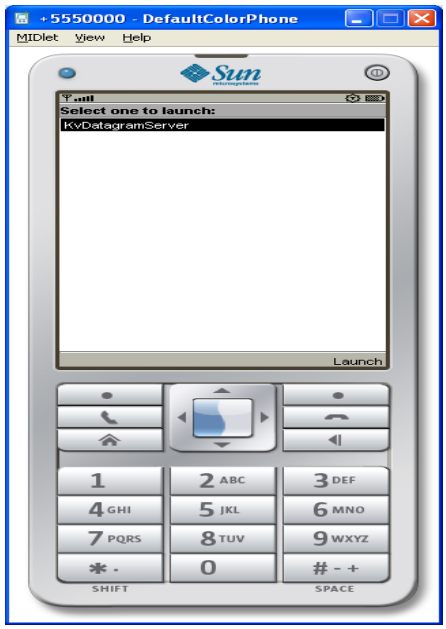


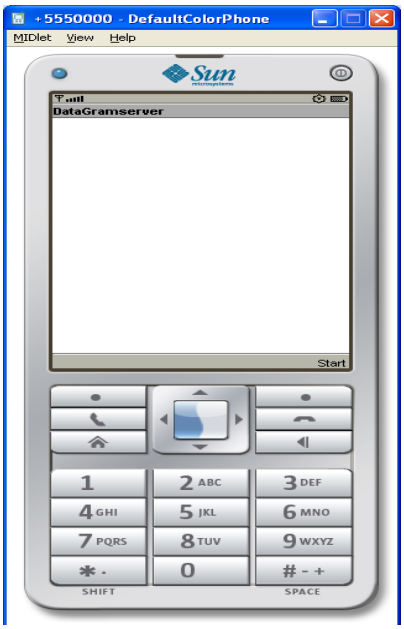


```
/*   Creating the datagram Server project   */

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
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(" ");
}
}
}
}
```

Output:





12. Login using Http Connection DATE:

- 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 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.

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import java.io.*;
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 commandActi on(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());
}
}
}
}
```

Output: