

Bhagwan Parshuram Institute Of Technology , Rohini

INDEX

Sr. No.	List of Experiment	Date of Experiment	Signature
1	To study about WML (Wireless Markup Language) Introduction to different tags in WML.		
2.	a) Write a WML program to print a Text on the mobile Screen b)Write a WML program to display your name in bold, italic, small, big and emphasize format.		
3.	a)Write a WML program to call 2 cards from same desk. b)Write a WML program to declare three cards and provide Hyper Link to move from one card to another card.		
4.	a)Write WML program to display table with three columns Image name, Image and third column contain hyperlink to open another card. b)Write WML program to display user personal information like Id, Name, Address, PhoneNo and pass same information on another deck.		
5.	a)Write a WML program to display username and password. b)Write a WML Program for creating a simple Option Groups. c)Having multiple fields and select one of them. d)Write a WML program to print user's special character.		

Sr. No.	List of Experiment	Date of Experiment	Signature
6.	a)Write a WML program to use the time control. b)Write a WML program to trigger On pick event.		
7.	a)Write a WML script to find absolute value with help of inbuilt function Lang.abs(). b) Write a WML script to find maximum out of two numbers with help of inbuilt function Lang.Max().		
8.	a)Perform WML Deck and WML Script program for Calculator. b)Perform WML Deck and WML Script program for currency conversion		
9.	a)Perform WML Deck and WML Script for date validation. b)Perform WML Deck and WML Script program for Email validation.		
10.	a)To install and configure Eclipse and JDK. b)Write an Android Program to display the text as toast on the mobile screen		
11.	a) Write an Android Program to design an application using Java Approach. b) Write an Android Program to design an application using XML Approach. c) Write an Android program to design an application using Hybrid Approach.		
12	Write a Program in NS3 to simulate DSDV		
13	Write a Program in NS3 to Simulate AODV		
14	Write a Program in NS3 to Simulate OLSR		

MOBILE COMPUTING

ETIT 452

Practical File



Submitted to:

Ms. Varsha Sharma

Submitted by :

Shivani Gautam

BTECH- IT (8TH Sem)

Roll. No. : 58

Roll No. : 35420803115

Bhagwan Parshuram Institute of Technology, PSP IV,
Sector – 17, Rohini, New Delhi – 110089

EXPERIMENT NO.1

INTRODUCTION

WML is a markup language designed especially for specifying and displaying content on WAP (Wireless Application Protocol) devices. WML is part of the WAP application environment, which requires the use of WML.

WML is the wireless equivalent of HTML for the Web. WML is based on XML and derived from xHTML (the XML version of HTML). There are many differences between WML and HTML. For example, WML has a different mechanism for linking between its pages called “cards” as compared to linking between HTML pages. WML browsers are stricter than HTML browsers by not being tolerant of errors. WML browsers enforce the WML requirement of matching closing “tags”, an XML characteristic.

WML works with the WAP micro browsers found on WAP devices. This browser is cognizant of the limited capabilities of WAP devices such as

- ☐ small displays,
- ☐ limited processing power,
- ☐ limited memory,
- ☐ narrow bandwidth connection,
- ☐ limited battery use without recharging

To address the limitations of WAP devices, WML uses the metaphor of card decks, and each page is referred to as a card. The card is the basic unit of navigation and user interface. The user can view only card at a time. WML browsers read the whole deck (complete document) from the server to minimize interaction with the server. Consequently, when flipping (navigating) between the cards in a deck, the browser does not contact the server. This eliminates delays (because each card contains very little text and users are likely to move quickly from one card to another).

A WAP deck is the equivalent of a Web page, the card being the portion of the Web page that can be seen on the screen. Navigation within the cards of a deck is done within the WAP device just as scrolling a Web page is done within the Web device. (without contacting servers in both cases).

An HTML writer does not worry about screen or display boundaries. Instead, the Web browser manages issues relating to the screen boundaries. But a WML writer must be aware of screen boundaries of WAP devices when writing code for cards.

Web server requests are routed through WAP gateways (proxies). A Web server may generate WML content for WAP devices or it may simply dish out HTML (XML).

In case the Web server generates HTML (XML), the WAP gateways must convert the HTML (XML) to WML. Before sending the WML to the WAP device, the gateway compresses it to WMLC (the C in WMLC is for compressed). If the WML generated by the Web sever is WMLC, then the compression step is skipped. Incidentally, converting HTML automatically to WML typically does not produce good or even usable results. For best results, Web servers should generate WML for WAP devices.

Advantages of WML

Like HTML, WML is easy to use. However, compared to HTML, WML has the following advantages in the context of wireless:

- ☐ WML is part of the WAP standard and its use is required.
- ☐ Transmission of WML (WMLC) documents requires less bandwidth compared to HTML documents because WML documents are simpler and WML is compressed before it is sent to the WAP device.
- ☐ Compared to HTML documents, displaying WML documents requires less processing power and memory. Consequently, a WAP device can work with a less powerful (cheaper) CPU and the use of less power means that the battery can operate longer without recharging.
- ☐ WML provides support for limited graphics with a limited gray scale.

Disadvantages of WML:

- ☐ Like HTML, WML does specify how the content is to be displayed. Thus micro browsers on different WAP devices are likely to display the WML content differently.
- ☐ WAP devices such as WAP phones will not accept large decks (1.4K for some WAP phones).
- There are many variations between WAP phones, for example Screen sizes, keypads, and soft keys can be different. Consequently, WML decks should be tested on at least the important WAP devices. This variation is similar to the variation found with Web browsers and their platforms. The problem is harder in case of WML because there are many more WAP devices than Web browsers and their platforms. Also, it is harder to figure out the “least common denominator”, i.e., set of features that will work reasonably well on all or most WAP devices.

Different Tags of WML are as follows:

Deck & Card Elements

WML Elements	Purpose
<code><!--></code>	Defines a WML comment
<code><wml></code>	Defines a WML deck (WML root)
<code><head></code>	Defines head information
<code><meta></code>	Defines meta information
<code><card></code>	Defines a card in a deck
<code><access></code>	Defines information about the access control of a deck

<template>	Defines a code template for all the cards in a deck
------------	---

Text Elements

WML Elements	Purpose
 	Defines a line break
<p>	Defines a paragraph
<table>	Defines a table
<td>	Defines a table cell (table data)
<tr>	Defines a table row
<pre>	Defines preformatted text

Text Formatting Tags

WML Elements	Purpose
	Defines bold text
<big>	Defines big text
	Defines emphasized text
<i>	Defines italic text
<small>	Defines small text

	Defines strong text
<u>	Defines underlined text

Image Elements

WML Elements	Purpose
	Defines an image

Anchor Elements

WML Elements	Purpose
<a>	Defines an anchor
<anchor>	Defines an anchor

Event Elements

WML Elements	Purpose
<do>	Defines a do event handler
<onevent>	Defines an onevent event handler
<postfield>	Defines a postfield event handler
<ontimer>	Defines an ontimer event handler
<onenterforward>	Defines an onenterforward handler
<onenterbackward>	Defines an onenterbackward handler
<onpick>	Defines an onpick event handler

Task Elements

WML Elements	Purpose
<code><go></code>	Represents the action of switching to a new card
<code><noop></code>	Says that nothing should be done
<code><prev></code>	Represents the action of going back to the previous card
<code><refresh></code>	Refreshes some specified card variables.

Input Elements

WML Elements	Purpose
<code><input></code>	Defines an input field
<code><select></code>	Defines a select group
<code><option></code>	Defines an option in a selectable list
<code><fieldset></code>	Defines a set of input fields
<code><optgroup></code>	Defines an option group in a selectable list

Variable Elements

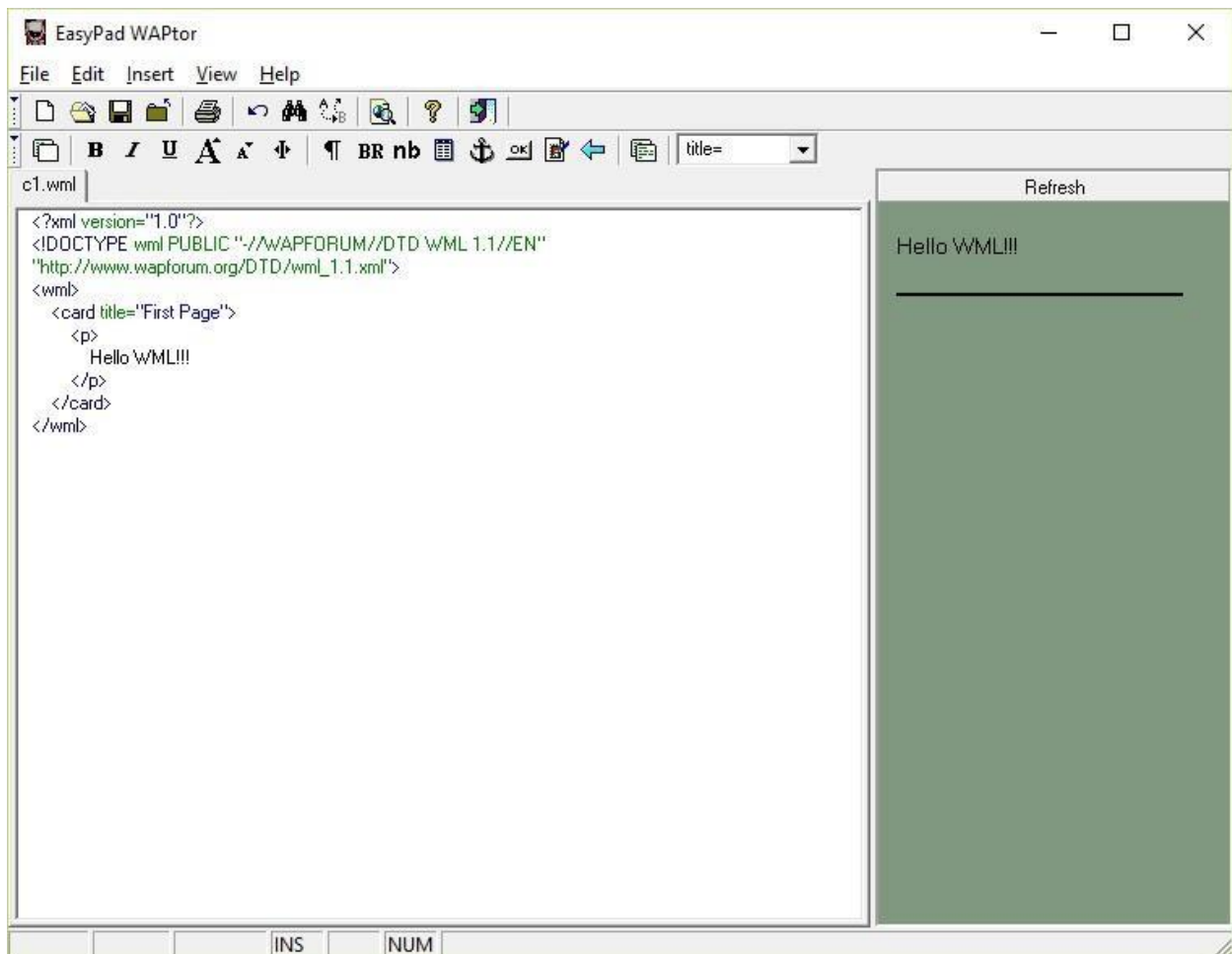
WML Elements	Purpose
<code><setvar></code>	Defines and sets a variable
<code><timer></code>	Defines a timer

EXPERIMENT NO. 2 (a)

Aim: Write a WML program to print a Text on the mobile Screen

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC
    "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<card title="First WML Example">
<p>Name</p><br>
<p>Roll Number</p><br>
</card>
</wml>
```

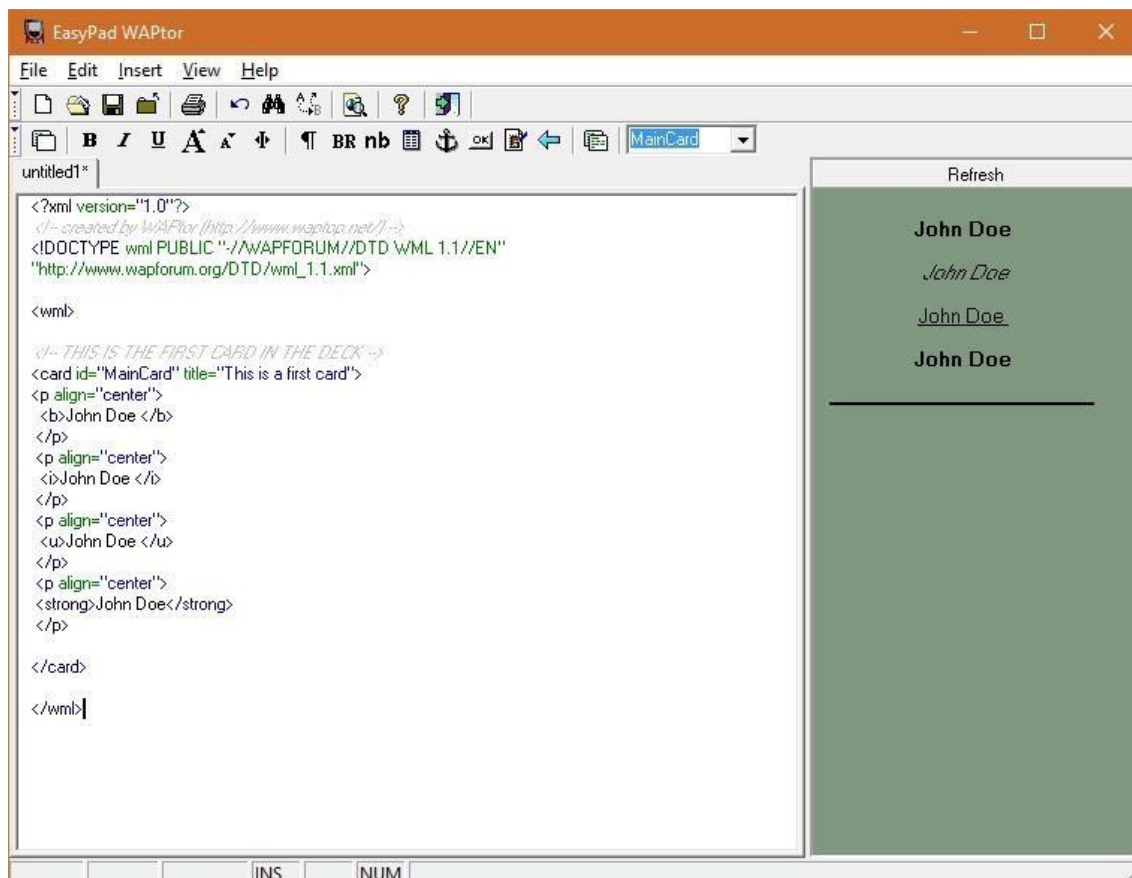


EXPERIMENT NO.2 (b)

Aim: Write a WML program to display your name in bold, italic, small, big and emphasize format.

```
<?xml version="1.0"?>
<!-- created by WAPtor (http://www.waptop.net/) -->
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<!-- THIS IS THE FIRST CARD IN THE DECK -->
<card id="MainCard" title="This is a first card">
<p align="center">
  <b>John Doe </b>
</p>
<p align="center">
  <i>John Doe </i>
</p>
<p align="center">
  <u>John Doe </u>
</p>
<p align="center">
  <strong>John Doe</strong>
</p>
</card>
</wml>
```



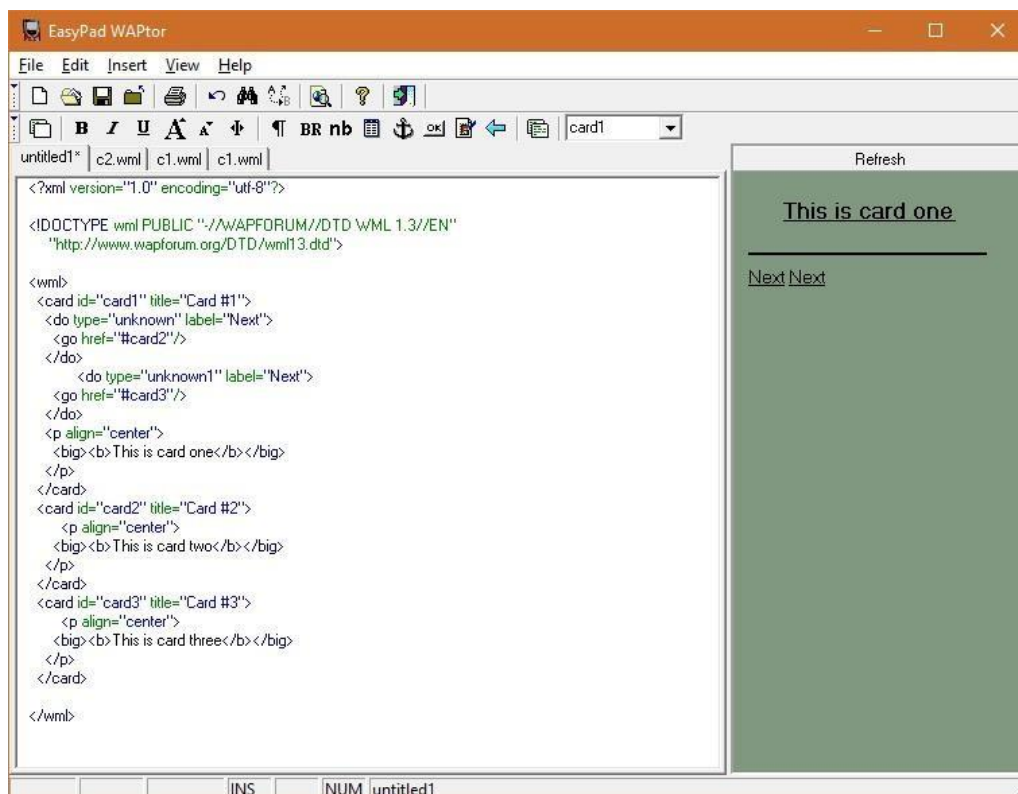
EXPERIMENT NO. 3 (a)

Aim: Write a WML program to call 2 cards from same desk.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
```

```
<wml>
<card id="card1" title="Card #1">
  <do type="unknown" label="Next">
    <go href="#card2"/>
  </do>
  <do type="unknown1" label="Next">
    <go href="#card3"/>
  </do>
  <p align="center">
    <big><b>This is card one</b></big>
  </p>
</card>
<card id="card2" title="Card #2">
  <p align="center">
    <big><b>This is card two</b></big>
  </p>
</card>
<card id="card3" title="Card #3">
  <p align="center">
    <big><b>This is card three</b></big>
  </p>
</card>
</wml>
```



EXPERIMENT NO. 3 (b)

Aim: Write a WML program to declare three cards and provide Hyper Link to move from one card to another card.

```
<?xml version="1.0" encoding="utf-8"?>

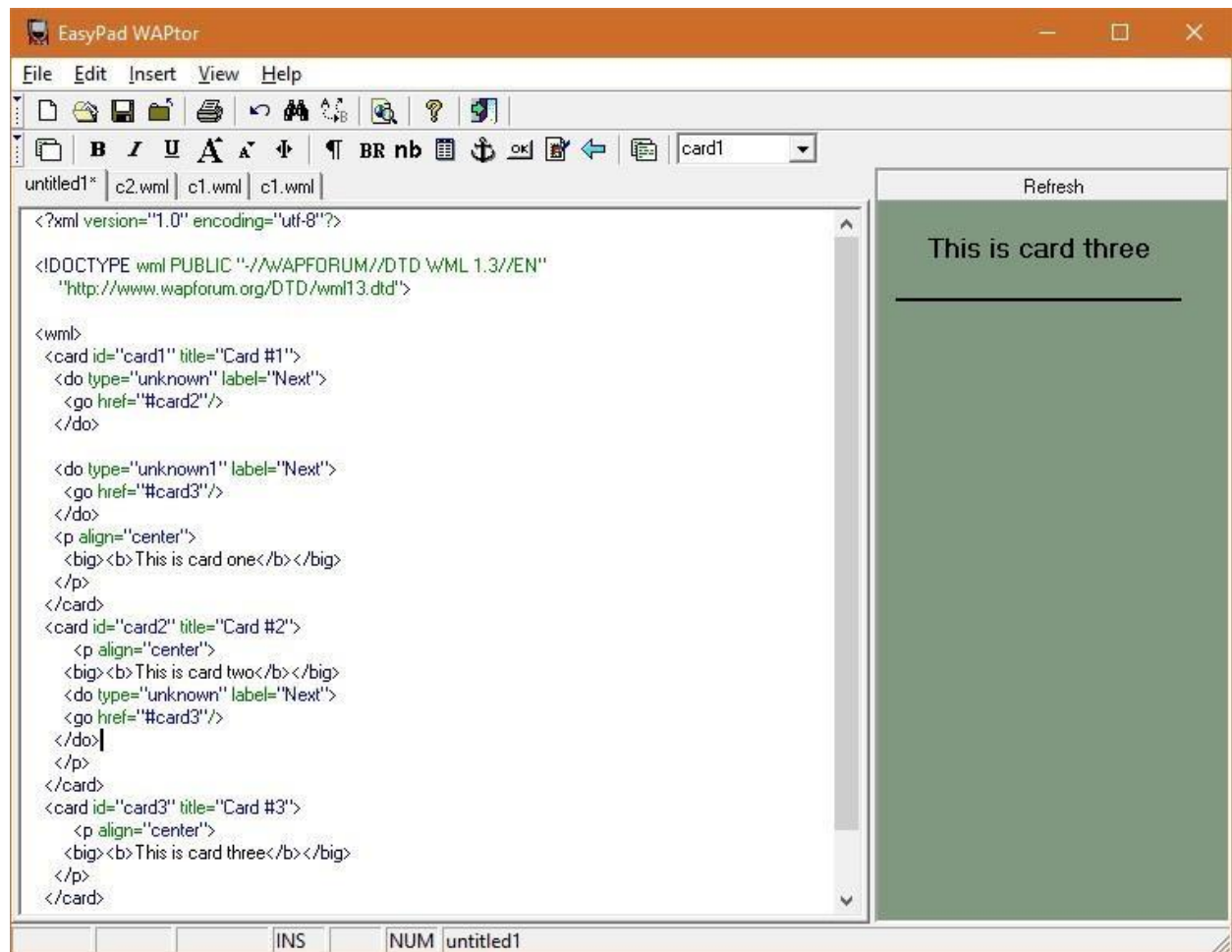
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
    "http://www.wapforum.org/DTD/wml13.dtd">

<wml>
  <card id="card1" title="Card #1">
    <do type="unknown" label="Next">
      <go href="#card2"/>
    </do>

    <do type="unknown1" label="Next">
      <go href="#card3"/>
    </do>

    <p align="center">
      <big><b>This is card one</b></big>
    </p>
  </card>
  <card id="card2" title="Card #2">
    <p align="center">
      <big><b>This is card two</b></big>
      <do type="unknown" label="Next">
        <go href="#card3"/>
      </do>
    </p>
  </card>
  <card id="card3" title="Card #3">
    <p align="center">
      <big><b>This is card three</b></big>
    </p>
  </card>

</wml>
```

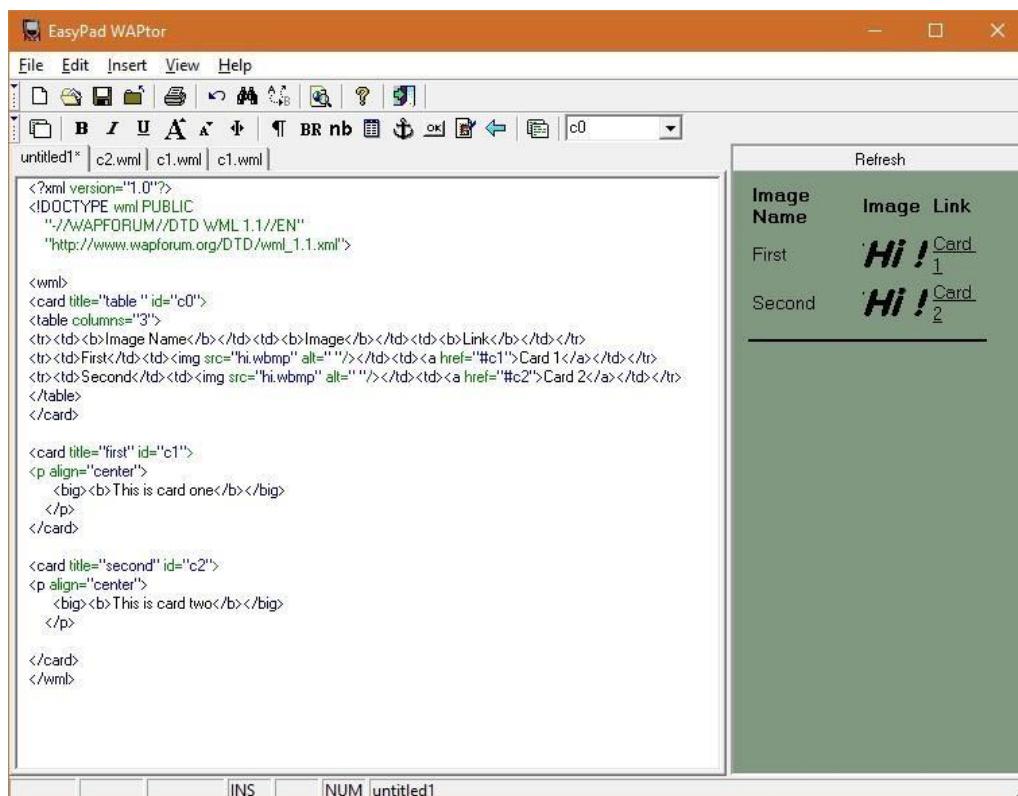


EXPERIMENT NO. 4 (a)

Aim: Write WML program to display table with three columns Image name, Image and third column contain hyperlink to open another card.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC
    "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">

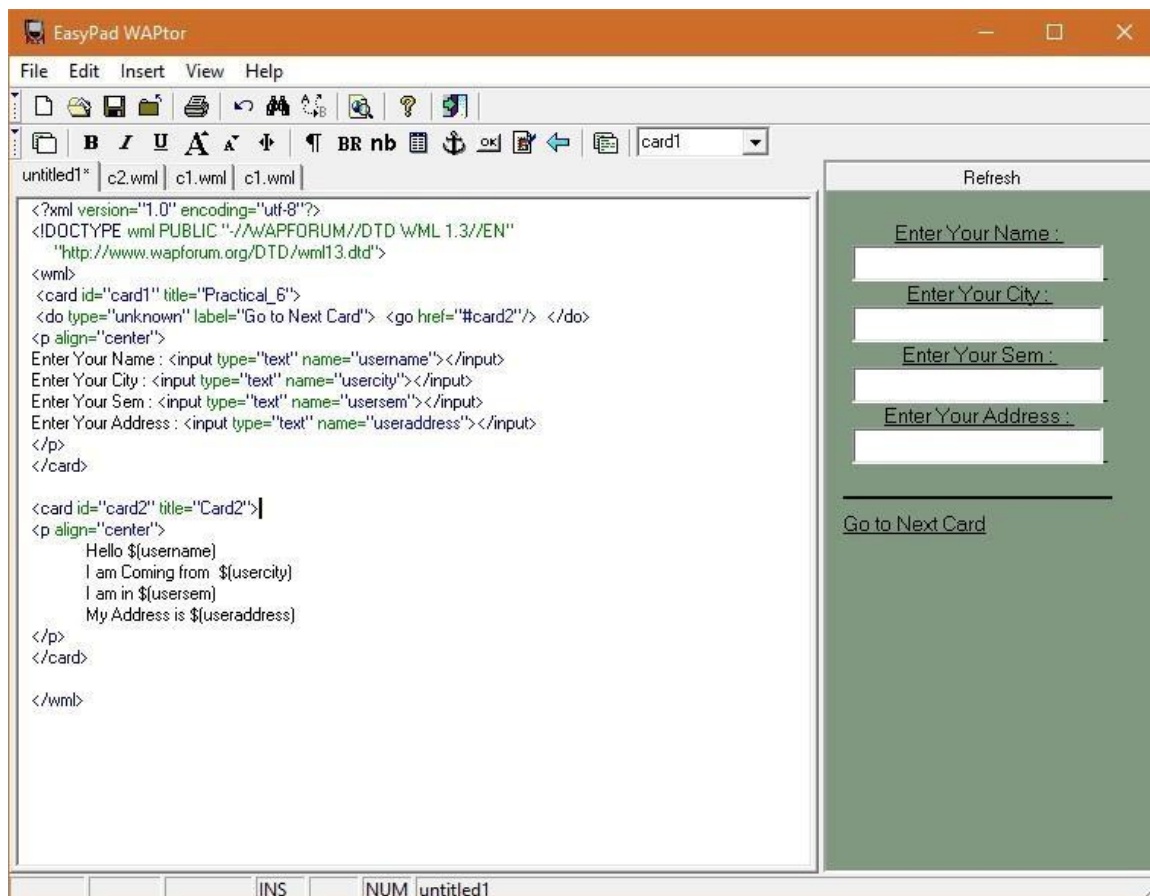
<wml>
<card title="table " id="c0">
<table columns="3">
<tr><td><b>Image Name</b></td><td><b>Image</b></td><td><b>Link</b></td></tr>
<tr><td>First</td><td></td><td><a href="#c1">Card
1</a></td></tr>
<tr><td>Second</td><td></td><td><a href="#c2">Card
2</a></td></tr>
</table>
</card>
<card title="first" id="c1">
<p align="center">
    <big><b>This is card one</b></big>
</p>
</card>
<card title="second" id="c2">
<p align="center">
    <big><b>This is card two</b></big>
</p>
</card>
</wml>
```



EXPERIMENT NO. 4 (b)

Aim: Write WML program to display user personal information like Id, Name, Address, Phone No and pass same information on another deck.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
    "http://www.wapforum.org/DTD/wml13.dtd">
<wml>
  <card id="card1" title="Practical_6">
    <do type="unknown" label="Go to Next Card"> <go href="#card2"/> </do>
    <p align="center">
      Enter Your Name : <input type="text" name="username"></input>
      Enter Your City : <input type="text" name="usercity"></input>
      Enter Your Sem : <input type="text" name="usersem"></input>
      Enter Your Address : <input type="text" name="useraddress"></input>
    </p>
  </card>
  <card id="card2" title="Card2">
    <p align="center">
      Hello $(username)
      I am Coming from $(usercity)
      I am in $(usersem)
      My Address is $(useraddress)
    </p>
  </card>
</wml>
```



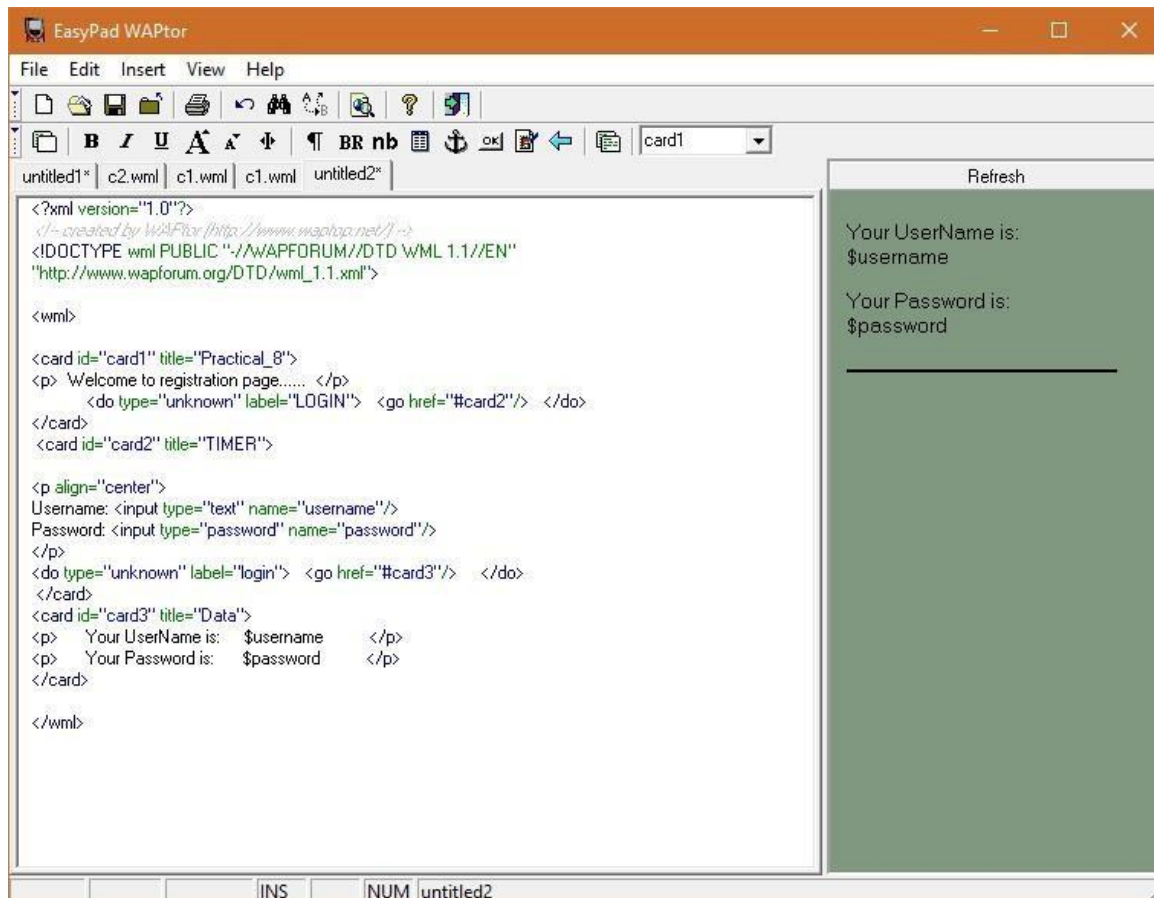
EXPERIMENT NO.5 (a)

Aim: Write a WML program to display username and password

```
<?xml version="1.0"?>
<!-- created by WAPtor (http://www.waptop.net/) -->
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<card id="card1" title="Practical_8">
<p> Welcome to registration page..... </p>
    <do type="unknown" label="LOGIN"> <go href="#card2"/> </do>
</card>
<card id="card2" title="TIMER">
<p align="center">
Username: <input type="text" name="username"/>
Password: <input type="password" name="password"/>
</p>
<do type="unknown" label="login"> <go href="#card3"/> </do>
</card>
<card id="card3" title="Data">
<p> Your UserName is: $username </p>
<p> Your Password is: $password </p>
</card></wml>
```



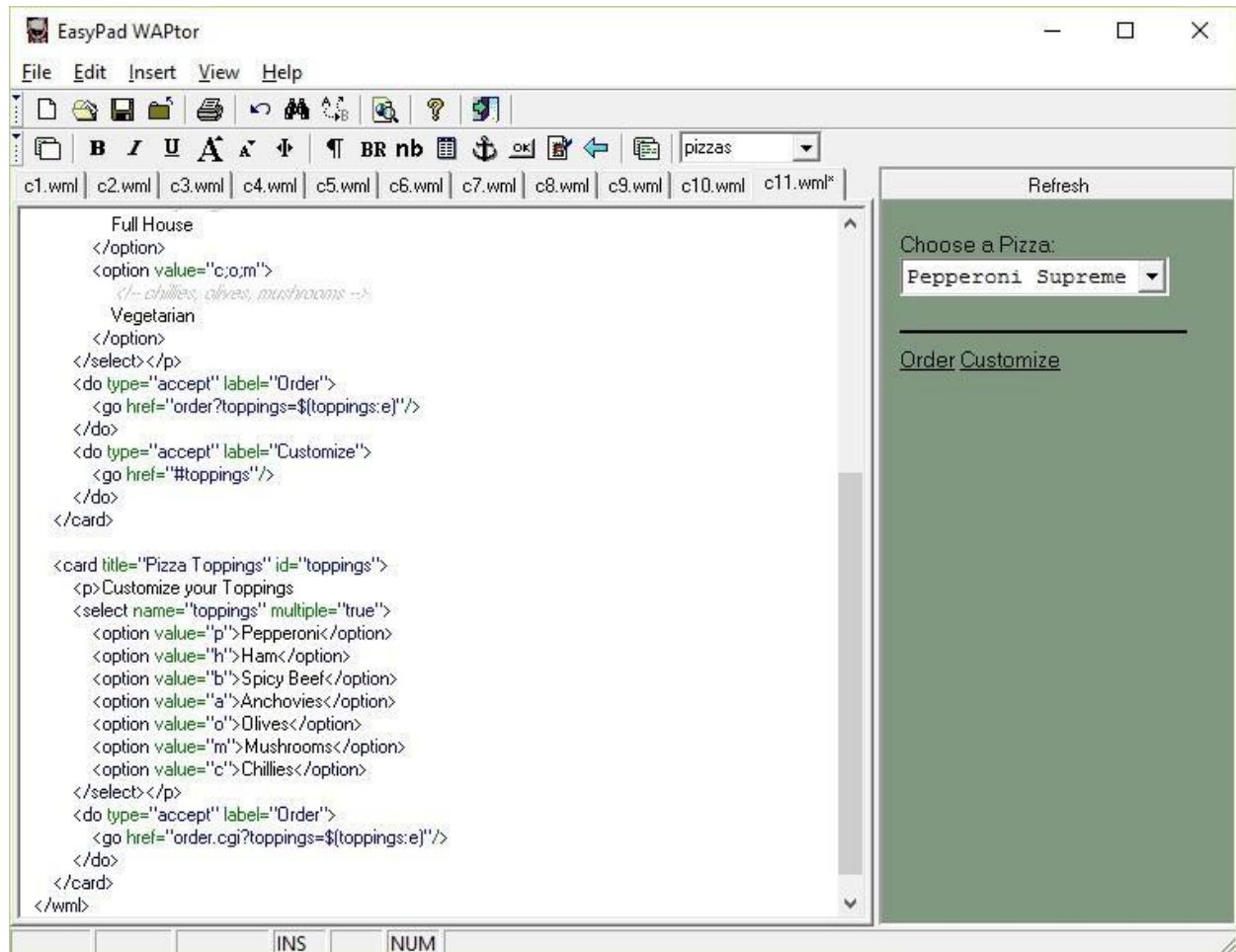
EXPERIMENT NO. 5 (b)

Aim: Write a WML Program for creating a simple Option Groups

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC
    "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
<card title="Set Pizzas" id="pizzas">
<p>Choose a Pizza:<select name="toppings">
<option value="">
<!-- no toppings -->
    Plain
</option>
<option value="p;m;o">
<!-- pepperoni, mushrooms, olives -->
    Pepperoni Supreme
</option>
<option value="p;b;h;a;m;o;c">
<!-- everything! -->
    Full House
</option>
<option value="c;o;m">
<!-- chillies, olives, mushrooms -->
    Vegetarian
</option>
</select></p>
<do type="accept" label="Order">
<go href="order?toppings=$(toppings:e)"/>
</do>
<do type="accept" label="Customize">
<go href="#toppings"/>
</do>
</card>
```

```
<card title="Pizza Toppings" id="toppings">
<p>Customize your Toppings
<select name="toppings" multiple="true">
<option value="p">Pepperoni</option>
<option value="h">Ham</option>
<option value="b">Spicy Beef</option>
<option value="a">Anchovies</option>
<option value="o">Olives</option>
<option value="m">Mushrooms</option>
<option value="c">Chillies</option>
</select></p>
<do type="accept" label="Order">
<go href="order.cgi?toppings=$(toppings:e)"/>
</do>
</card>
</wml>
```

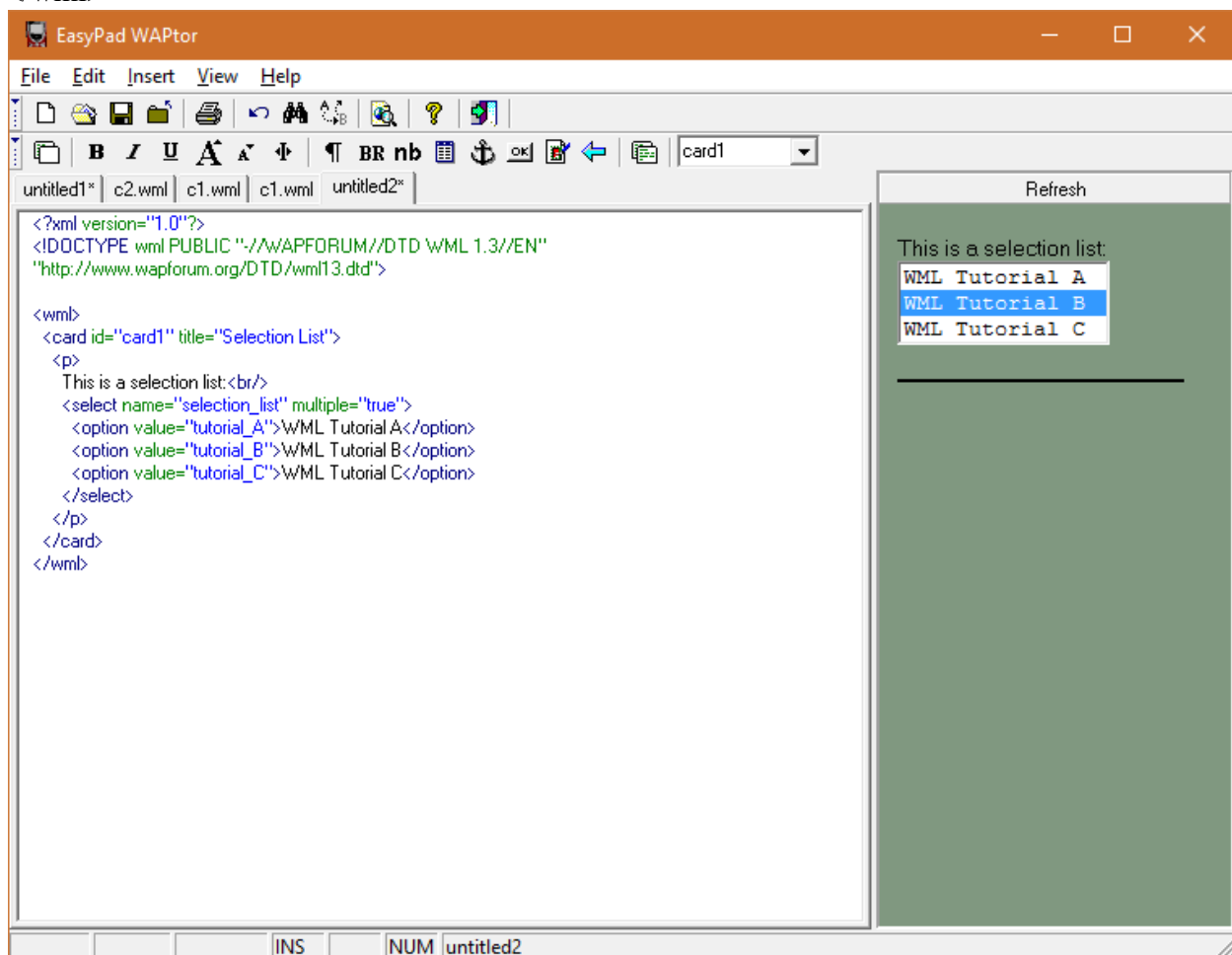


EXPERIMENT NO. 5 (c)

Aim: Write a WML program having multiple fields and select one of them

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
```

```
<wml>
  <card id="card1" title="Selection List">
    <p>
      This is a selection list:<br/>
      <select name="selection_list" multiple="true">
        <option value="tutorial_A">WML Tutorial A</option>
        <option value="tutorial_B">WML Tutorial B</option>
        <option value="tutorial_C">WML Tutorial C</option>
      </select>
    </p>
  </card>
</wml>
```



EXPERIMENT NO. 6 (a)

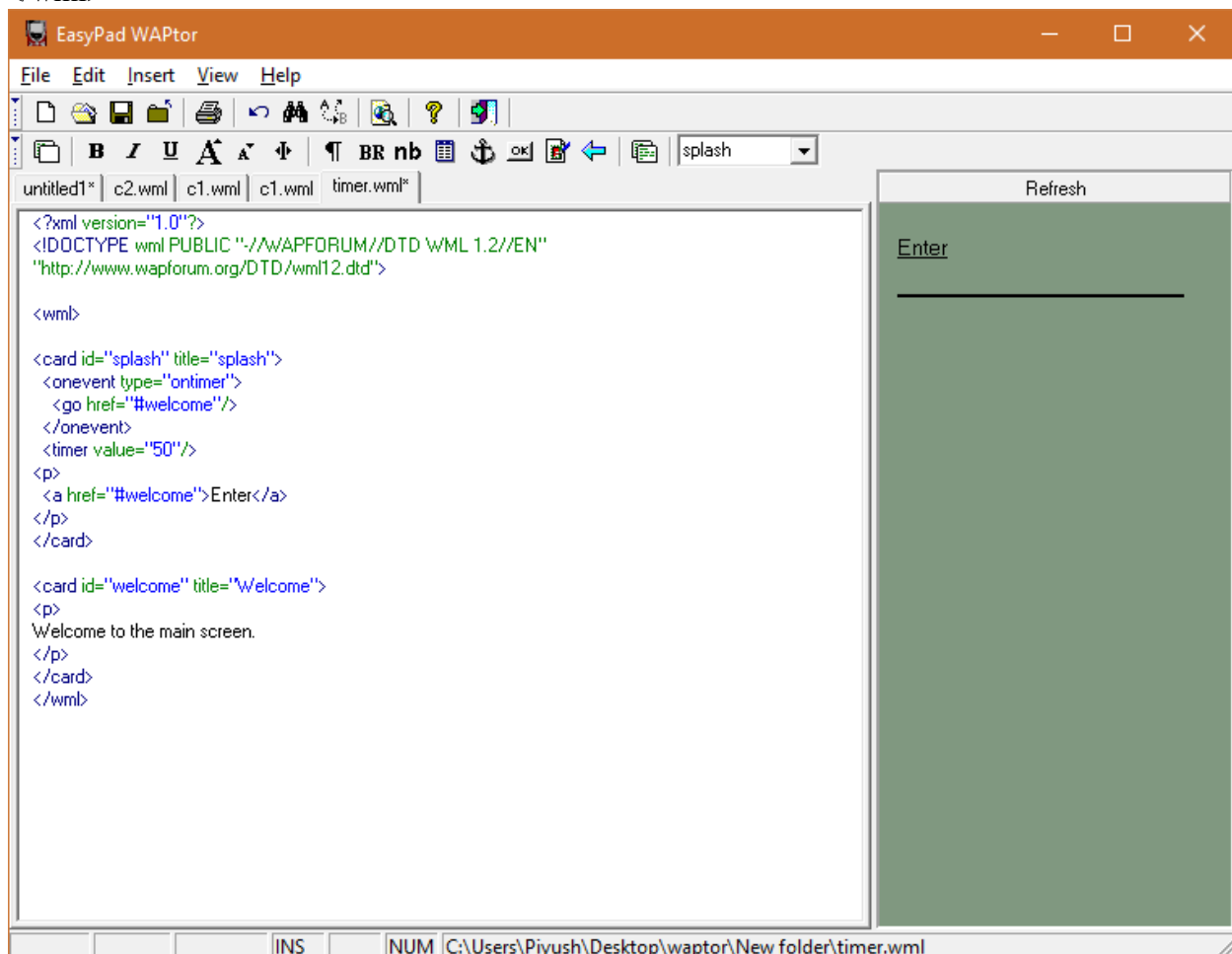
Aim: Write a WML program to use the time control.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>

<card id="splash" title="splash">
  <onevent type="ontimer">
    <go href="#welcome"/>
  </onevent>
  <timer value="50"/>
<p>
  <a href="#welcome">Enter</a>
</p>
</card>

<card id="welcome" title="Welcome">
<p>
Welcome to the main screen.
</p>
</card>
</wml>
```



EXPERIMENT NO. 6 (b)

Aim: Write a WML program to trigger On pick event

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">

<wml>
  <card id="tutorial_tc" title="WML Tutorial">
    <p>
      Hello, welcome to our WML tutorial.<br/><br/>
      Table of Contents:<br/>

      <select name="selection_list">
        <option onpick="#tutorial_ch1">Chapter 1</option>
        <option onpick="#tutorial_ch2">Chapter 2</option>
        <option>
          <onevent type="onpick">
            <go href="#tutorial_ch3"/>
          </onevent>
          Chapter 3
        </option>
        <option>
          <onevent type="onpick">
            <go href="#tutorial_ch4"/>
          </onevent>
          Chapter 4
        </option>
      </select>
    </p>
  </card>

  <card id="tutorial_ch1" title="WML Tutorial Ch1">
    <p>
      <em>Chapter 1: WML Introduction</em><br/><br/>
      ...
    </p>
  </card>

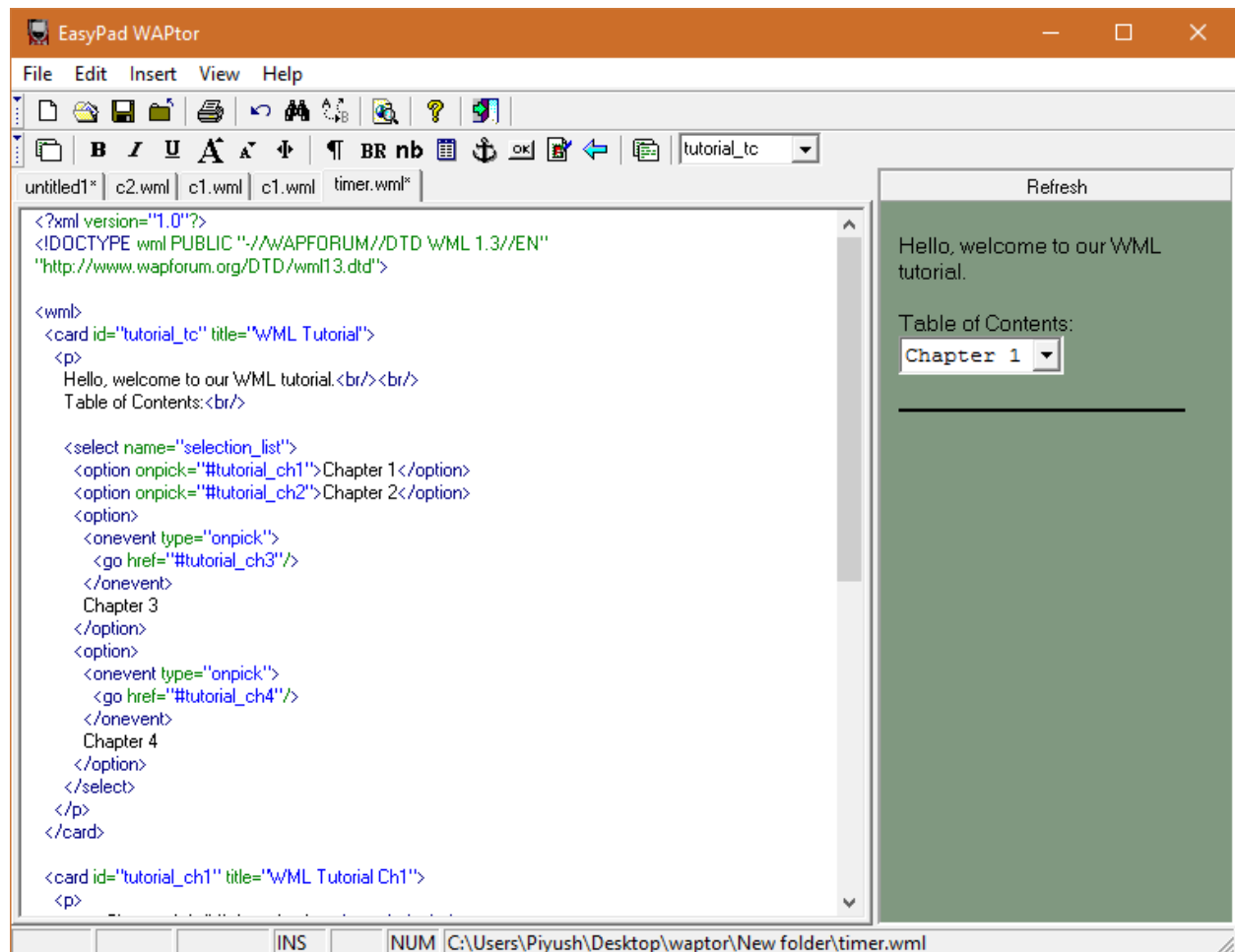
  <card id="tutorial_ch2" title="WML Tutorial Ch2">
    <p>
      <em>Chapter 2: WML Deck and Card</em><br/><br/>
      ...
    </p>
  </card>

  <card id="tutorial_ch3" title="WML Tutorial Ch3">
    <p>
      <em>Chapter 3: WML Document Structure</em><br/><br/>
      ...
    </p>
  </card>
```

```

<card id="tutorial_ch4" title="WML Tutorial Ch4">
  <p>
    <em>Chapter 4: WML Generic Metadata</em><br/><br/>
    ...
  </p>
</card>
</wml>

```



EXPERIMENT NO. 7(a)

Aim: Write a WML script to find absolute value with help of inbuilt function

Lang.abs(). abs.wml

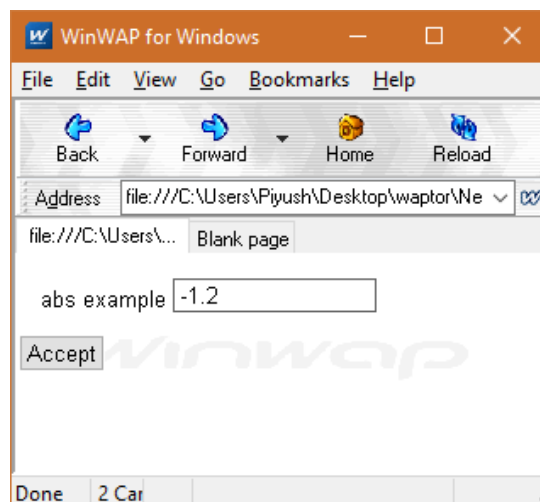
```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.WAPforum.org/DTD/wml_1.1.xml">
<wml>
```

```
<card id="card1">
  <p>
    Enter value <input type="text" name="num"></input>
  </p>
  <do type="accept">
    <go href="abs.wmls#findabs('$(num)')"/>
  </do>
</card>
```

```
<card id="card2">
  <p>
    original number = $(number)
    <br />
    absolute number = $(absnumber)
  </p>
</card>
</wml>
```

abs.wmls

```
extern function findabs(n)
{
  var absnum = Lang.abs(n);
  WMLBrowser.setVar("number", n);
  WMLBrowser.setVar("absnumber", absnum);
  WMLBrowser.go("abs.wml#card2");
};
```



EXPERIMENT NO. 7(b)

Aim: Write a WML script to find maximum out of two numbers with help of inbuilt function Lang.Max().

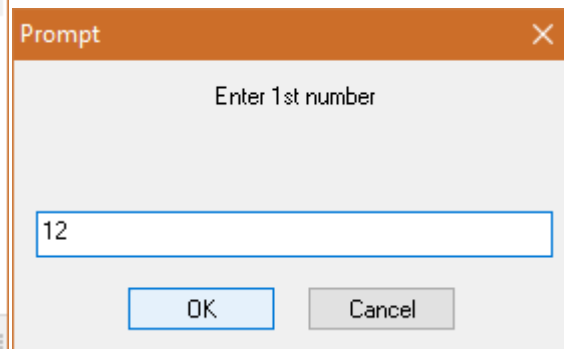
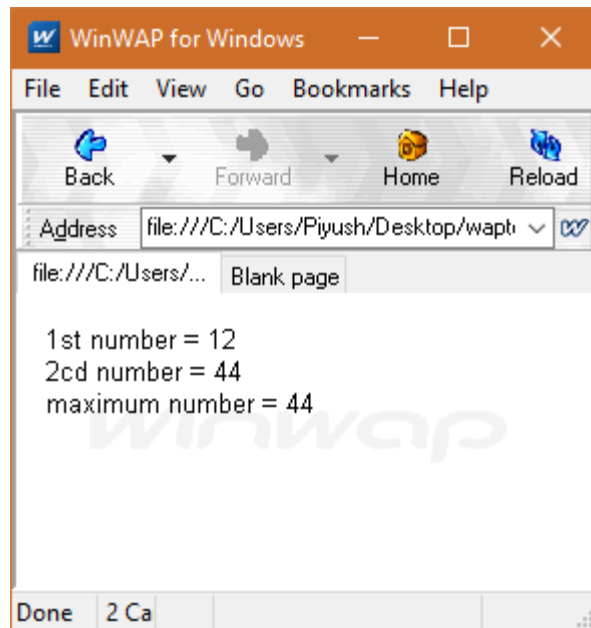
minmax.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.WAPforum.org/DTD/wml_1.1.xml">
<wml>

<card id="card1">
  <p>
    max example
  </p>
  <do type="accept">
    <go href="minmax.wmls#findmax()" />
  </do>
</card>
<card id="card2">
  <p>
    1st number = $(number1)
    <br />
    2nd number = $(number2)
    <br />
    maximum number = $(maxnumber)
  </p>
</card>
</wml>
```

minmax.wmls

```
extern function findmax()
{
  var result1 = Dialogs.prompt("Enter 1st number", "");
  var result2 = Dialogs.prompt("Enter 2cd number", "");
  var maxnum = Lang.max(result1, result2);
  WMLBrowser.setVar("number1", result1);
  WMLBrowser.setVar("number2", result2);
  WMLBrowser.setVar("maxnumber", maxnum);
  WMLBrowser.go("minmax.wml#card2");
};
```

EXPERIMENT NO. 8 (a)

Aim: Perform WML Deck and WML Script program for Calculator

calc.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC
    "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
  <card title="Calculator" newcontext="true">
    <!-- Initialize the result variable. -->
    <onevent type="onenterforward">
      <refresh>
        <setvar name="display" value="0.0"/>
        <setvar name="number" value=""/>
        <setvar name="register" value=""/>
        <setvar name="lastop" value=""/>
      </refresh>
    </onevent>

    <!-- Result display. -->
    <p>$(display)</p>

    <!-- Rows of keys. -->
    <p>
      <a href="calc.wmls#digit(7)">7</a>
      <a href="calc.wmls#digit(8)">8</a>
      <a href="calc.wmls#digit(9)">9</a>
      <a href="calc.wmls#op('+)">+</a>
      <br/>
      <a href="calc.wmls#digit(4)">4</a>
      <a href="calc.wmls#digit(5)">5</a>
      <a href="calc.wmls#digit(6)">6</a>
      <a href="calc.wmls#op('-)">-</a>
      <br/>
      <a href="calc.wmls#digit(1)">1</a>
      <a href="calc.wmls#digit(2)">2</a>
      <a href="calc.wmls#digit(3)">3</a>
      <a href="calc.wmls#op('*)">*</a>
      <br/>
      <a href="calc.wmls#digit(0)">0</a>
      <a href="calc.wmls#point( )">.</a>
      <a href="calc.wmls#op('=)">=</a>
      <a href="calc.wmls#op('/')>/</a>
      <br/>
```

```

    <anchor>C <!-- Clear resets number being entered. -->
      <refresh>
        <setvar name="display" value="0.0"/>
        <setvar name="number" value=""/>
      </refresh>
    </anchor>
    <anchor>AC <!-- All Clear resets everything. -->
      <refresh>
        <setvar name="display" value="0.0"/>
        <setvar name="number" value=""/>
        <setvar name="register" value=""/>
        <setvar name="lastop" value=""/>
      </refresh>
    </anchor>
  </p>
</card>
</wml>

```

calc.wmls

```

extern function digit (d)
{
  /* Read the current number as a string from the browser context. */
  var number = WMLBrowser.getVar ("number");

  /* Add digit to number. (Note that the variable 'number' actually
   * contains a string at this point, so this concatenates strings.) */
  number += d;

  /* Set browser variables and refresh the display. */
  WMLBrowser.setVar ("number", number);
  set_display (Lang.parseFloat ("0" + number));
}

/*
 * Add a decimal point to the number currently being entered.
 */
extern function point ( )
{
  /* Read the current number as a string from the browser context. */
  var number = WMLBrowser.getVar ("number");

  /* Ignore the key if there's already a decimal point. */
  if (String.find (number, '.') >= 0)
    return;

  /* Add a decimal point to the number. */
  number += '.';
}

```

```

/* Set browser variables and refresh the display. */
WMLBrowser.setVar ("number", number);
set_display (Lang.parseFloat ("0" + number));
}

/*
 * Handle an operator key: perform the last operation and store this
 * operator until the next number has been entered.
 */
extern function op (op)
{
    /* Fetch the register and convert to floating point. */
    var register = Lang.parseFloat (WMLBrowser.getVar ("register"));

    /* Fetch the number and convert to floating point. */
    var number = Lang.parseFloat (WMLBrowser.getVar ("display"));

    /* Fetch the last operator key. */
    var lastop = WMLBrowser.getVar ("lastop");

    /* Work out what operation needs to be performed and perform it. */
    if (lastop == 'add')
        register += number;
    else if (lastop == 'sub')
        register -= number;
    else if (lastop == 'mul')
        register *= number;
    else if (lastop == 'div')
        register /= number;
    else
        register = number;

    /* Store the new operator for next time. */
    WMLBrowser.setVar ("lastop", op);

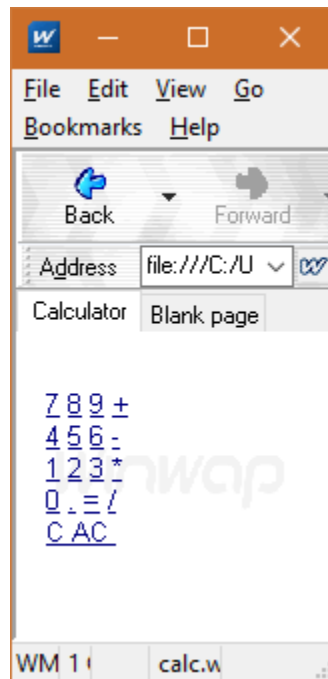
    /* Clear the number so we can enter a new one. */
    WMLBrowser.setVar ("number", "");

    /* Both the display and the register are the result of the operation. */
    WMLBrowser.setVar ("register", String.toString (register));
    set_display (register);
}

/*
 * Set the display browser variable and refresh the display.
 */
function set_display (display)
{
    /* Handle an invalid calculation result. */
    if (!isvalid display)
        display = "(error)";
}

```

```
/* Set the browser variable. */  
WMLBrowser.setVar ("display", display);  
  
/* Refresh the display. */  
WMLBrowser.refresh ( );  
  
}
```



EXPERIMENT NO. 8 (b)

Aim: Perform WML Deck and WML Script program for currency conversion

currency.wml

```
<wml>

<card id="card1" title="Currency" newcontext="true">
<p>
Amount: <input format="N*M" name="input" title="Enter Amt.:"/>

From: <select name="from" value="RUP" title="From:">
<option value="RUP">Indian Rupees</option>
<option value="AUS">Australian Dollar</option>
<option value="USD">US Dollar</option>
</select>

To: <select name="to" value="USD" title="To:">
<option value="RUP">Indian Rupees</option>
<option value="AUS">Australian Dollar</option>
<option value="USD">US Dollar</option>
</select>

<br/> = $(ans)

<do type="accept" label="Calculate">
<go href="currency.wmls#convert('ans','$(from)','$(to)','$(input))"/>
</do>

</p>
</card>
</wml>
```

currency.wmls

```
extern function convert(ans,from,to,input)
{

var mult = 0.0;
var rString = "Hello";
var re;

if (from == "RUP")
{

var RUP_USD = 0.02222;
var RUP_AUS = 0.06667;
```

```

if (to == "RUP")
mult = 1.0;
else if (to == "USD")
mult = RUP_USD;
else if (to == "AUS")
mult = RUP_AUS;
}
else if (from == "USD")
{

var USD_RUP = 45;
var USD_AUS = 3;

if (to == "USD")
mult = 1.0;
else if (to == "RUP")
mult = USD_RUP;
else if (to == "AUS")
mult = USD_AUS;

}
else if (from == "AUS")
{

var AUS_RUP = 15;
var AUS_USD = 0.3334;

if (to == "AUS")
mult = 1.0;
else if (to == "USD")
mult = AUS_USD;
else if (to == "RUP")
mult = AUS_RUP;
}

if (mult != 0.0) {

re = input * mult;
rString = String.toString(re);
rString = String.format("%.2f", rString);
}

WMLBrowser.setVar(ans,rString);
WMLBrowser.refresh();
}

```

W C _ □ ×

File Edit View Go
Bookmarks Help

Back Forward

Address file:///C:/User W

Currency Blank page

Amount:
100

From:
Indian Rupees

To:
US Dollar

= Calculate

Don 1 (

W C _ □ ×

File Edit View Go
Bookmarks Help

Back Forward

Address file:///C:/User W

Currency Blank page

Amount:
2.2

From:
Indian Rupees

To:
US Dollar

= Calculate

Don 1 (

EXPERIMENT NO. 9 (a)

Aim: Perform WML Deck and WML Script for date

validation date.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.WAPforum.org/DTD/wml_1.1.xml">
<wml>

<card id="card1">
  <p>
    Enter a date: <br>
    <input type="text" name="date"></input>
  </p>

  <do type="accept" label="Go">
<go href="date.wmls#checkDate($(date))"/>
</do>
</card>

<card id="card2">
<p>The date is $(Valid)</p>
</card>

</wml>
```

date.wmls

```
extern function checkDate(date)
{
  var status=isDateValid(date);
  WMLBrowser.setVar("Valid",status);
  WMLBrowser.go("date.wml#card2");
}

extern function isDateValid(date)
{
  var mm = String.subString(date, 0, 2);
  var dd = String.subString(date, 2, 2);
  var yyyy = String.subString(date, 4, 4);

  mm = Lang.parseInt(mm);
  dd = Lang.parseInt(dd);
  yyyy = Lang.parseInt(yyyy);

  if (mm<1 || mm>12)
    return false;
```

```

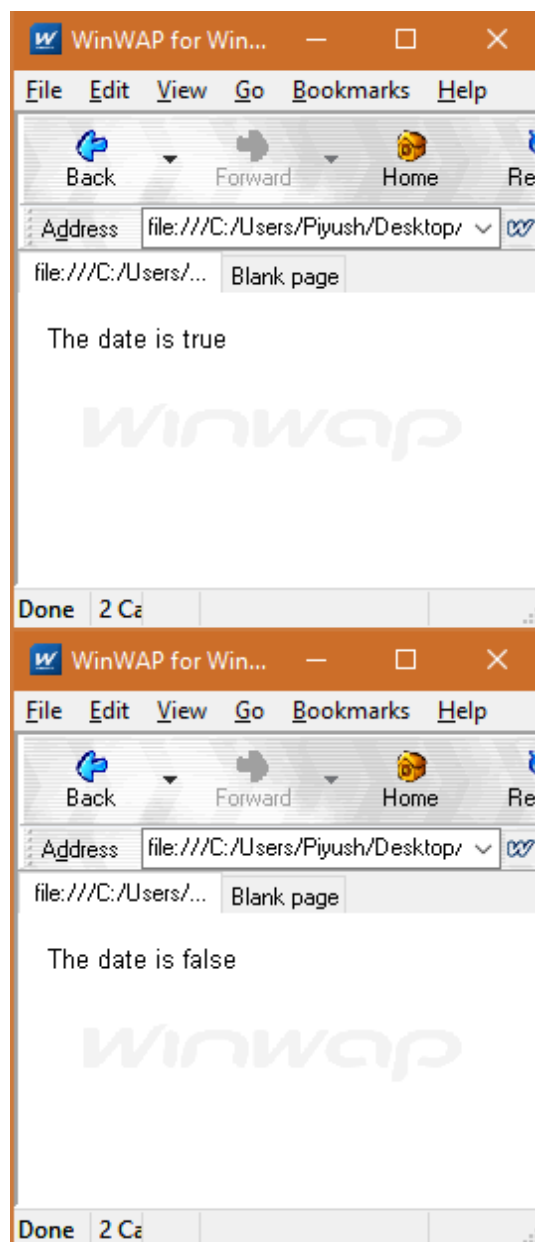
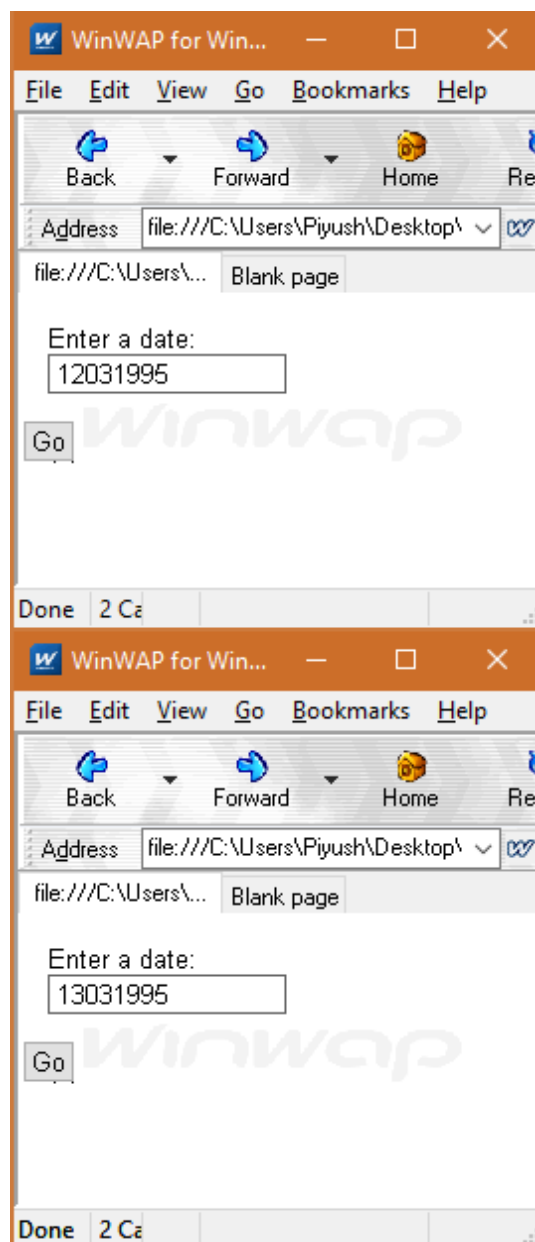
var maxDay = 31;
if (4==mm || 6==mm || 9==mm || 11==mm)
    maxDay = 30;
if (2==mm){
    if (0 == yyyy%4)

        maxDay = 29; // Leap year
    else
        maxDay = 28;
}

if (dd<1 || dd>maxDay)
    return false;

return true;

```



EXPERIMENT NO. 9 (b)

Aim : WML Deck and WML Script program for Email

validation email.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.WAPforum.org/DTD/wml_1.1.xml">
<wml>

<card id="card1">
  <p>
    Enter an Email: <br>
    <input type="text" name="mail"></input>
  </p>

  <do type="accept" label="Go">
<go href="email.wmls#checkEmail($(mail))"/>
</do>
</card>

<card id="card2">
<p>The Email is $(Valid)</p>
</card>

</wml>
```

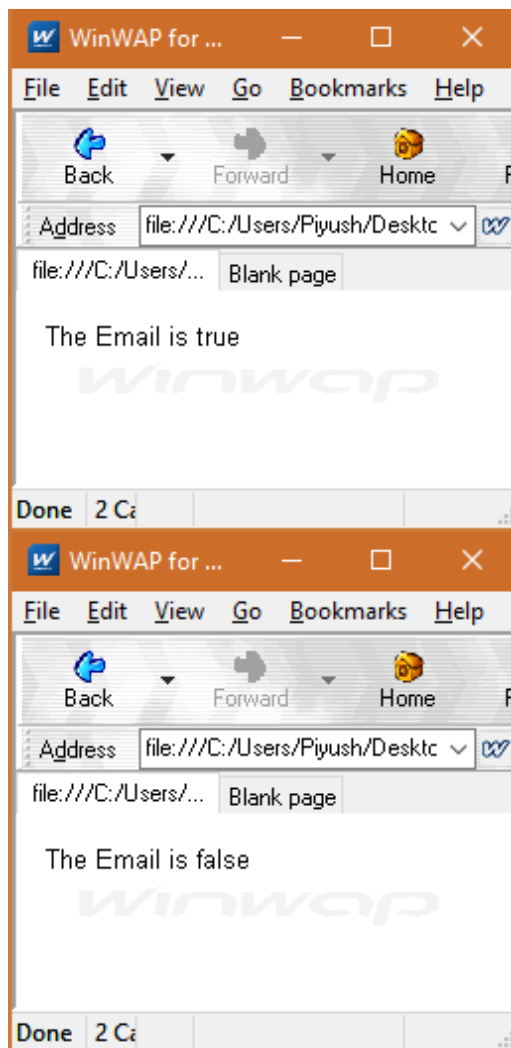
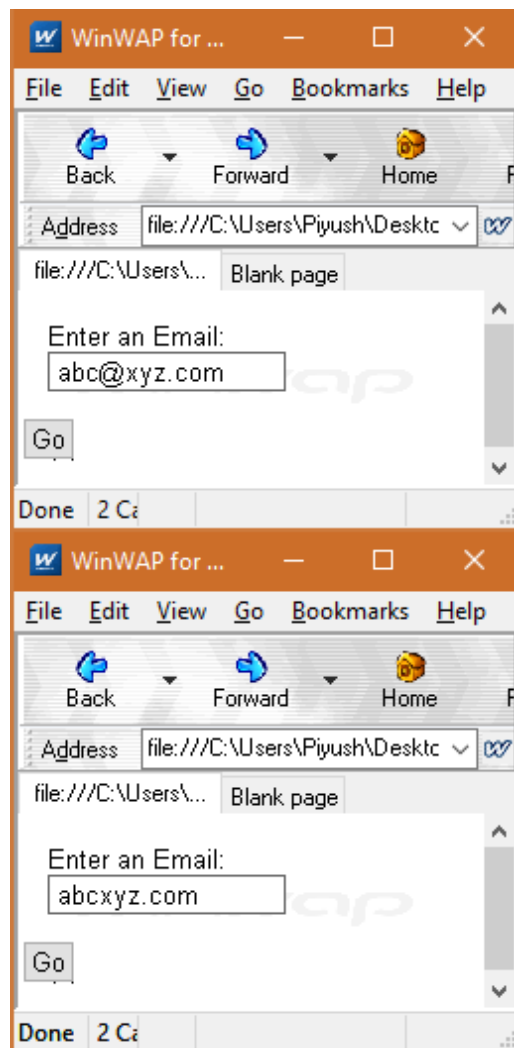
email.wmls

```
extern function checkEmail(mail)
{
  var status=isEmailValid(mail);
  WMLBrowser.setVar("Valid",status);
  WMLBrowser.go("email.wml#card2");
}

extern function isEmailValid(mail)
{
  if (String.elements(mail, "@") != 2)
    return false;

  var element_1 = String.elementAt(mail, 0, "@");
  var element_2 = String.elementAt(mail, 1, "@");
  if (""==element_1 || ""==element_2)
    return false;

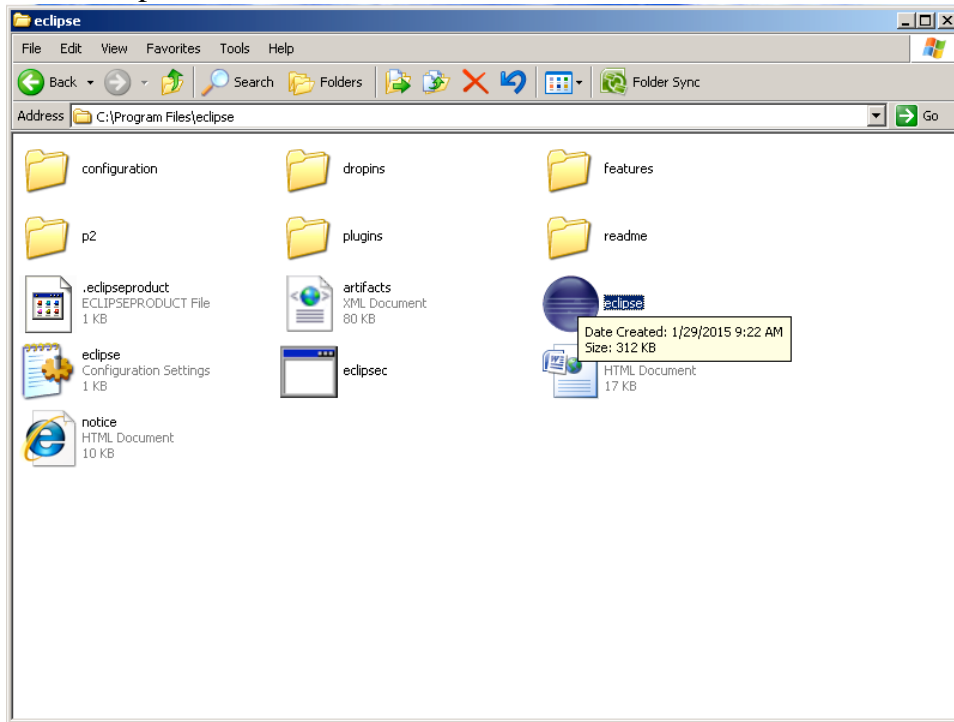
  return true;
}
```



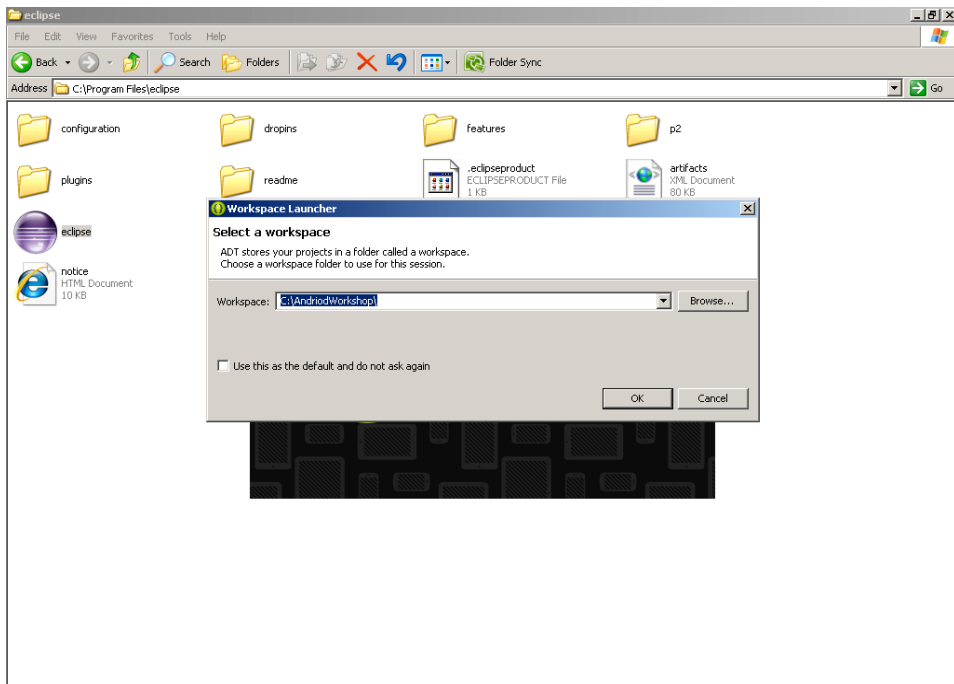
EXPERIMENT NO. 10 a

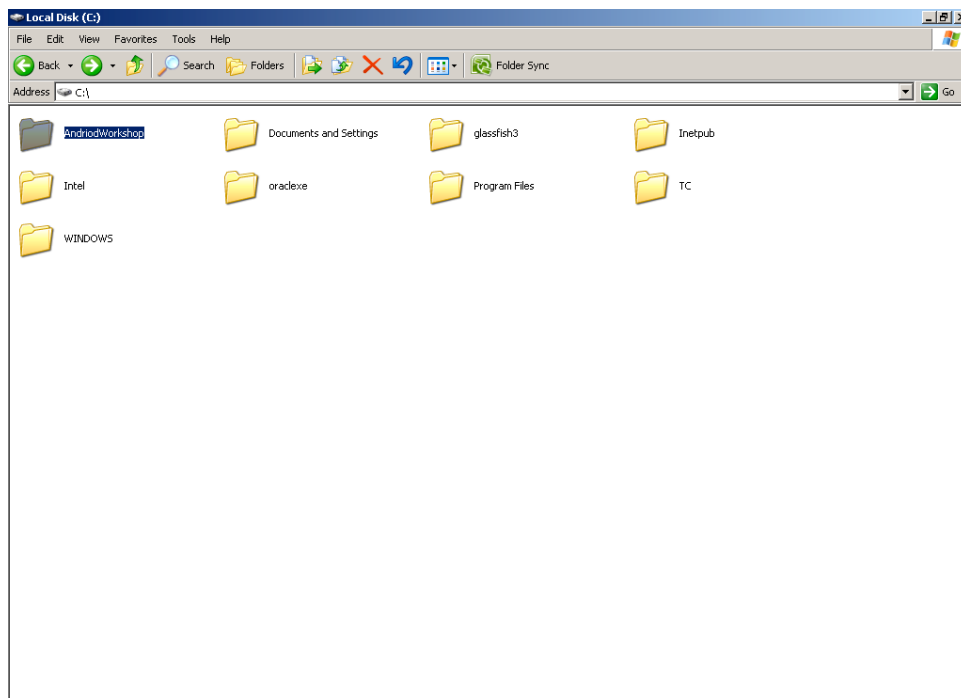
Aim: To install and configure Eclipse and JDK. Steps:

1. Go to eclipse and run the executable file.

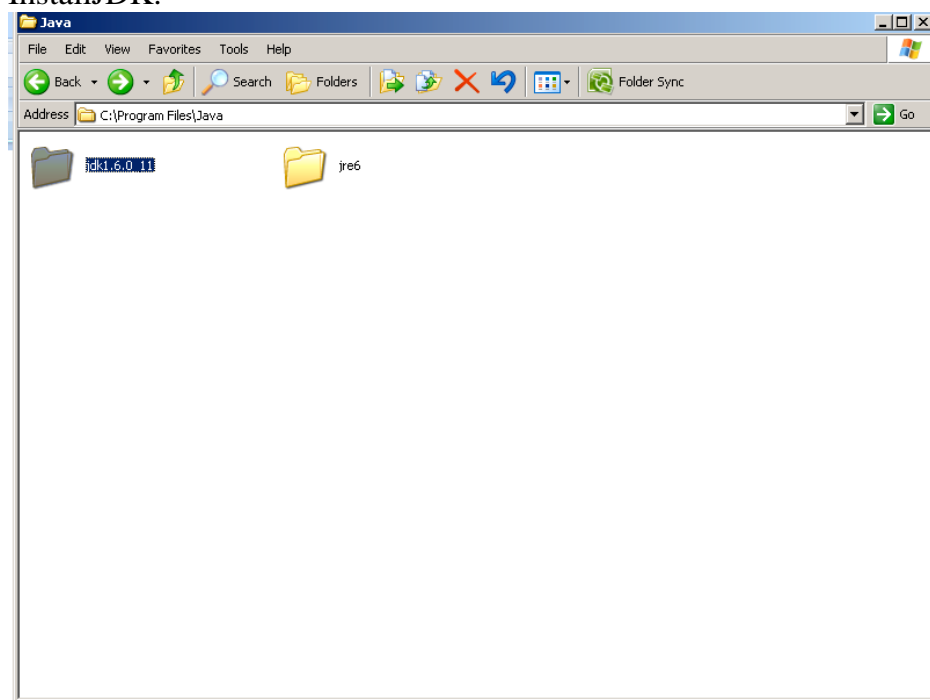


2. Make a folder in C drive.

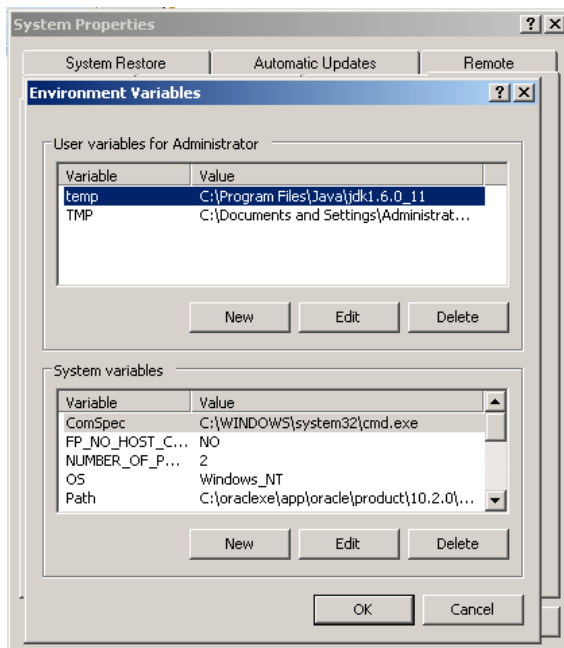




3. InstallJDK.



4. Set Environmental Variable.



EXPERIMENT NO. 10 b

Aim: Write a program to display the text as toast on the mobile screen.

Source Code:

```
package com.example.hybridapproach;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import
android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class HybridActivity extends Activity implements

    OnClickListener { Button b1;
    Button b2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hybrid);

        // Refer XML Controls in Java

        b1 = (Button)
        findViewById(R.id.button1); b2 =
        (Button) findViewById(R.id.button2);
        b1.setOnClickListener(this);
        b2.setOnClickListener(this);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.hybrid, menu);
        return true;
    }

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        switch(v.getId()){
```



```

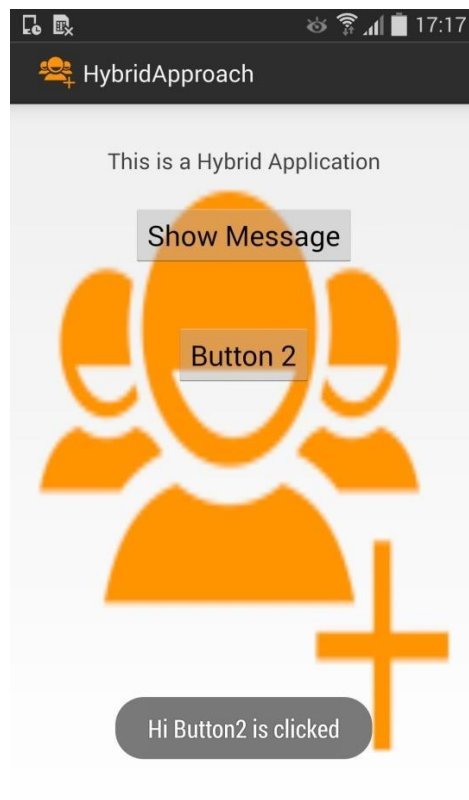
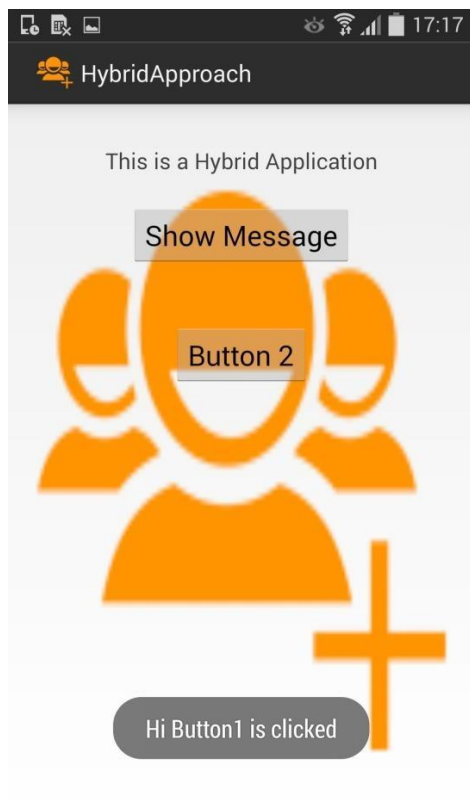
        case R.id.button1:
            Toast.makeText(this, "Hi Button1 is clicked",
                Toast.LENGTH_SHORT).show();
            break;
        case
            R.id.button2:
            Toast.makeText(this, "Hi Button2 is clicked",
                Toast.LENGTH_SHORT).show();
            break;
    }

}

}

```

Output:



EXPERIMENT NO. 11 a

Aim: Write a program to design an application using Java Approach.

Source Code:

```
package
com.example.javaapproach;
import android.os.Bundle;
import
android.app.Activity;
import
android.view.Menu;
import
android.view.View;
import
android.view.View.OnClickListener;
import android.widget.Button;
import
android.widget.EditText;
import
android.widget.LinearLayout;
import
android.widget.TextView;
import android.widget.Toast;

public class JavaApproachActivity extends Activity implements

    OnClickListener { EditText et1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //setContentView(R.layout.activity_java_approach);

        LinearLayout layout = new LinearLayout(this);
        layout.setOrientation(LinearLayout.VERTICAL);
        setTitle("Java Approach App");

        TextView tv = new TextView(this);
        tv.setText("This is our Java Approach
        Application"); layout.addView(tv);

        et1 = new
        EditText(this);
        et1.setHint("Enter
        Message");
```

```

        layout.addView(et1);

        Button b1 = new
        Button(this);
        b1.setText("Show
        Message");
        b1.setOnClickListener(th
        is); b1.setId(1);
        layout.addView(b1);

        Button b2 = new
        Button(this);
        b2.setText("Show HI");
        b2.setOnClickListener(this);
        b2.setId(2);
        layout.addView(b2);

        setContentView(layout);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.java_approach, menu);
        return true;
    }

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        switch(v.getId()){

            case 1: String msg = et1.getText().toString();
                    Toast.makeText(this, msg,
                    Toast.LENGTH_SHORT).show(); break;

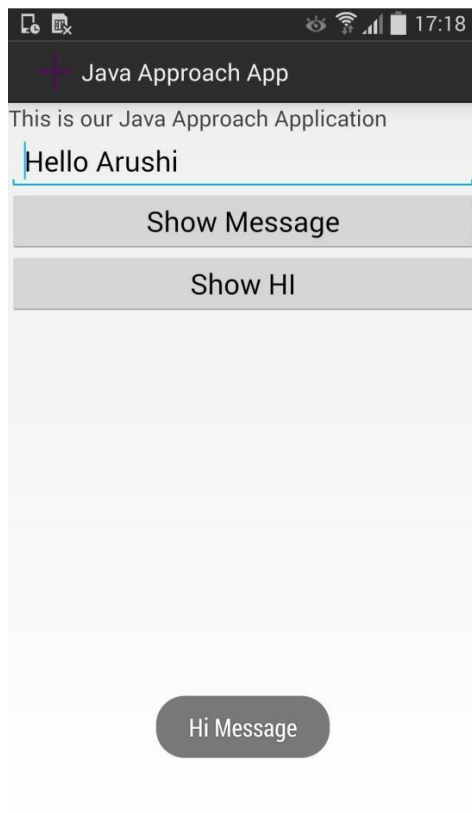
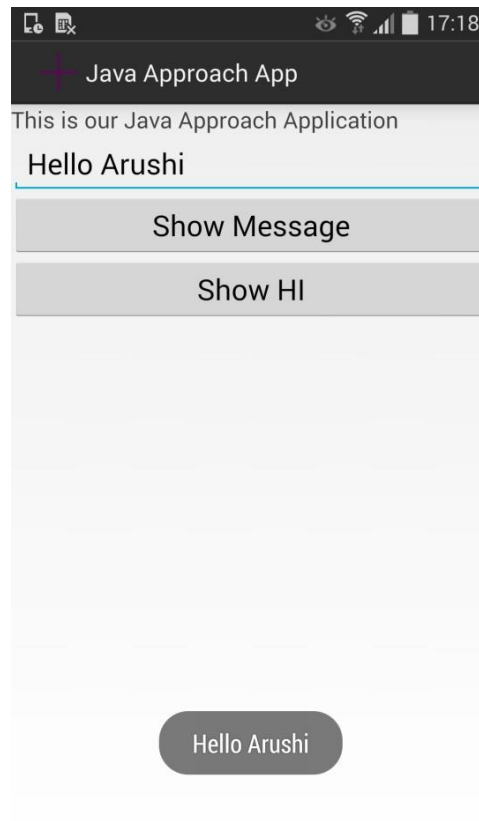
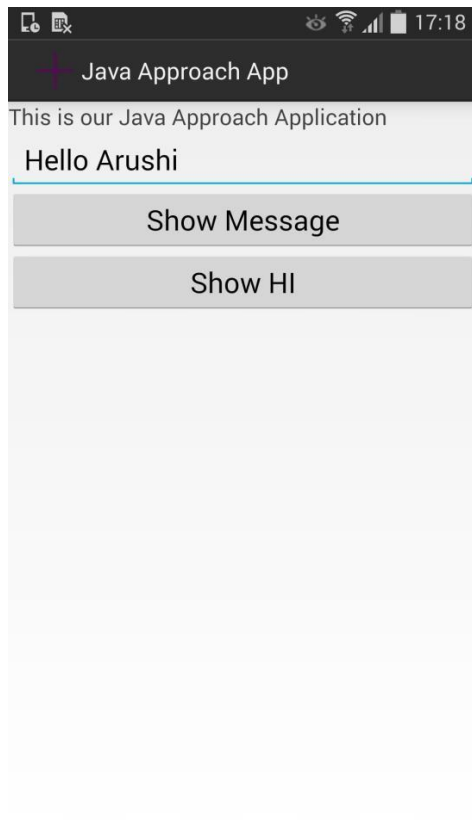
            case 2: Toast.makeText(this, "Hi Message",
                    Toast.LENGTH_SHORT).show(); break;
        }

    }

}

```

Output:



EXPERIMENT NO. 11 b

Aim: Write a program to design an application using XML Approach.

Source Code:

```
package
com.example.xmlapproach; import
android.os.Bundle;
import android.app.Activity;
import android.telephony.SmsManager;
import android.view.Menu;
import android.view.View;
import
android.widget.EditText;
import android.widget.Toast;

public class XmlApproachActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_xml_approach);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.xml_approach, menu);
        return true;
    }

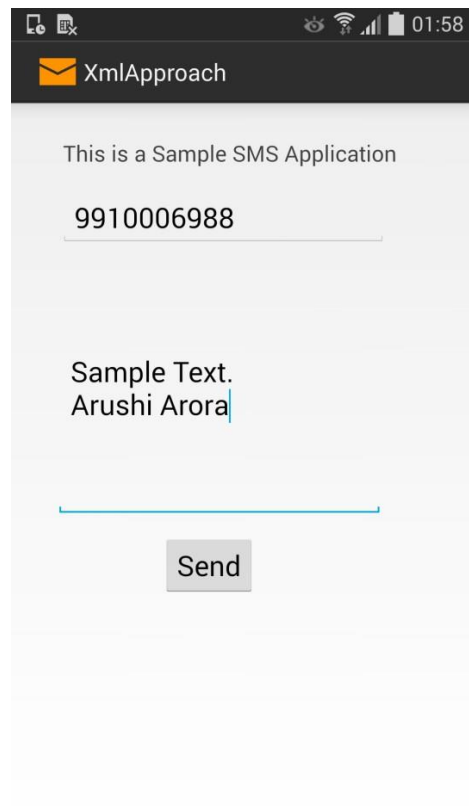
    public void sendSMS(View v){

        EditText et1 =
        (EditText)findViewById(R.id.editText1); EditText et2
        = (EditText)findViewById(R.id.editText2);

        String dst = et1.getText().toString();
        String msg = et2.getText().toString();

        SmsManager sm = SmsManager.getDefault();
        sm.sendTextMessage(dst, null, msg, null, null);
        Toast.makeText(this, "Sms Sent", Toast.LENGTH_SHORT).show();
    }
}
```

Output:



The screenshot shows an Android application interface with a dark header bar containing the title 'XmlApproach' and an envelope icon. The main content area is light gray and contains the text 'This is a Sample SMS Application'. Below this is a text input field with the number '9910006988'. Further down is another text input field containing 'Sample Text.' and 'Arushi Arora'. A blue underline is visible below the second input field. At the bottom of the form is a gray button labeled 'Send'.

01:58

XmlApproach

This is a Sample SMS Application

9910006988

Sample Text.
Arushi Arora

Send

EXPERIMENT NO. 11 c

Aim: Write a program to design an application using Hybrid Approach.

Source Code:

```
package com.example.hybridapproach;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import
android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class HybridActivity extends Activity implements

    OnClickListener { Button b1;
    Button b2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hybrid);

        // Refer XML Controls in Java

        b1 = (Button)
        findViewById(R.id.button1); b2 =
        (Button) findViewById(R.id.button2);
        b1.setOnClickListener(this);
        b2.setOnClickListener(this);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.hybrid, menu);
        return true;
    }

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        switch(v.getId()){
```

```

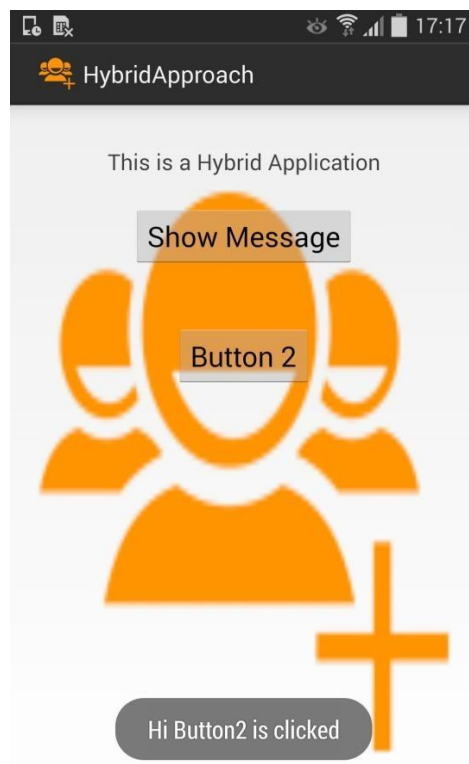
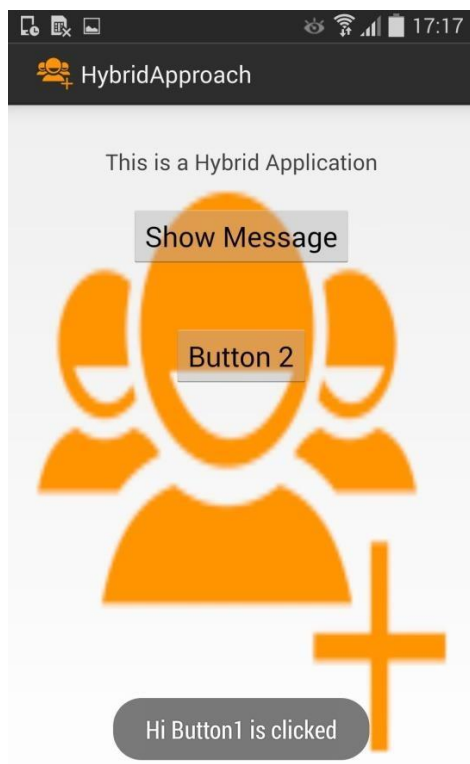
        case R.id.button1:
            Toast.makeText(this, "Hi Button1 is clicked",
                Toast.LENGTH_SHORT).show();
            break;
        case
            R.id.button2:
            Toast.makeText(this, "Hi Button2 is clicked",
                Toast.LENGTH_SHORT).show();
            break;
    }

}

}

```

Output:



EXPERIMENT NO. 12

Aim: Write a Program in NS3 to simulate DSDV

Source Code:

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/applications-module.h" #include "ns3/mobility-module.h" #include "ns3/config-store-
module.h" #include "ns3/wifi-module.h"
#include "ns3/internet-module.h" #include "ns3/dsdv-helper.h" #include <iostream>
#include <cmath>

using namespace ns3;

uint16_t port = 9;

NS_LOG_COMPONENT_DEFINE ("DsdvManetExample");

class DsdvManetExample
{
public:
DsdvManetExample ();
void CaseRun (uint32_t nWifis, uint32_t nSinks,
double totalTime, std::string rate, std::string phyMode, uint32_t nodeSpeed,
uint32_t periodicUpdateInterval, uint32_t settlingTime,
double dataStart,
bool printRoutes std::string CSVfileName);

private:
uint32_t m_nWifis; uint32_t m_nSinks; double m_totalTime; std::string m_rate; std::string m_phyMode;
uint32_t m_nodeSpeed;
uint32_t m_periodicUpdateInterval; uint32_t m_settlingTime;
double m_dataStart; uint32_t bytesTotal; uint32_t packetsReceived; bool m_printRoutes; std::string
m_CSVfileName;

NodeContainer nodes; NetDeviceContainer devices; Ipv4InterfaceContainer interfaces;

private:
void CreateNodes ();
void CreateDevices (std::string tr_name); void InstallInternetStack (std::string tr_name); void
InstallApplications ();
void SetupMobility ();
void ReceivePacket (Ptr <Socket> );
Ptr <Socket> SetupPacketReceive (Ipv4Address, Ptr <Node> ); void CheckThroughput ();

};
```

```

int main (int argc, char **argv)
{
DsdvManetExample test; uint32_t nWifis = 30; uint32_t nSinks = 10; double totalTime = 100.0;
std::string rate
("8kbps");
std::string phyMode ("DsssRate11Mbps"); uint32_t nodeSpeed = 10; // in m/s std::string appl = "all";
uint32_t periodicUpdateInterval = 15; uint32_t settlingTime = 6;
double dataStart = 50.0;
bool printRoutingTable = true;
std::string CSVfileName = "DsdvManetExample.csv";

CommandLine cmd;
cmd.AddValue ("nWifis", "Number of wifi nodes[Default:30]", nWifis); cmd.AddValue ("nSinks",
"Number of wifi
sink nodes[Default:10]", nSinks); cmd.AddValue ("totalTime", "Total Simulation time[Default:100]",
totalTime);
cmd.AddValue ("phyMode", "Wifi Phy mode[Default:DsssRate11Mbps]", phyMode); cmd.AddValue
("rate", "CBR traffic rate[Default:8kbps]", rate);
cmd.AddValue ("nodeSpeed", "Node speed in RandomWayPoint model[Default:10]", nodeSpeed);
cmd.AddValue
("periodicUpdateInterval", "Periodic Interval Time[Default=15]", periodicUpdateInterval); cmd.AddValue
("settlingTime", "Settling Time before sending out an update for changed metric[Default=6]",
settlingTime);
cmd.AddValue ("dataStart", "Time at which nodes start to transmit data[Default=50.0]", dataStart);
cmd.AddValue
("printRoutingTable", "print routing table for nodes[Default:1]", printRoutingTable);

cmd.AddValue ("CSVfileName", "The name of the CSV output file
name[Default:DsdvManetExample.csv]",
CSVfileName);
cmd.Parse (argc, argv);

std::ofstream out (CSVfileName.c_str ()); out << "SimulationSecond," << "ReceiveRate," <<
"PacketsReceived," << "NumberOfSinks," << std::endl;
out.close ();

SeedManager::SetSeed (12345);

Config::SetDefault ("ns3::OnOffApplication::PacketSize", StringValue ("1000"));
Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue (rate));
Config::SetDefault ("ns3::WifiRemoteStationManager::NonUnicastMode", StringValue (phyMode));
Config::SetDefault ("ns3::WifiRemoteStationManager::RtsCtsThreshold", StringValue ("2000"));

test = DsdvManetExample ();
test.CaseRun (nWifis, nSinks, totalTime, rate, phyMode, nodeSpeed, periodicUpdateInterval, settlingTime,
dataStart,
printRoutingTable, CSVfileName);

return 0;
}

```

```

DsdvManetExample::DsdvManetExample ()
: bytesTotal (0),
  packetsReceived (0)
{
}

void
DsdvManetExample::ReceivePacket (Ptr <Socket> socket)
{
  NS_LOG_UNCOND (Simulator::Now ().GetSeconds () <<" Received one packet!");
  Ptr <Packet> packet;
  while ((packet = socket->Recv ()))
  {
    bytesTotal += packet->GetSize (); packetsReceived += 1;
  }
}

void DsdvManetExample::CheckThroughput ()
{
  double kbs = (bytesTotal * 8.0) / 1000; bytesTotal = 0;

  std::ofstream out (m_CSVfileName.c_str (), std::ios::app);

  cout << (Simulator::Now ().GetSeconds () <<" ,"<< kbs <<" ,"<< packetsReceived <<" ,"<< m_nSinks <<
  std::endl;
  out.close (); packetsReceived = 0;
  Simulator::Schedule (Seconds (1.0), &DsdvManetExample::CheckThroughput, this);
}

Ptr <Socket>
DsdvManetExample::SetupPacketReceive (Ipv4Address addr, Ptr <Node> node)
{
  TypeId tid = TypeId::LookupByName ("ns3::UdpSocketFactory"); Ptr <Socket> sink =
  Socket::CreateSocket (node, tid); InetSocketAddress local = InetSocketAddress (addr, port);
  sink->Bind (local);
  sink->SetRecvCallback (MakeCallback ( &DsdvManetExample::ReceivePacket, this));

  return sink;
}

void
DsdvManetExample::CaseRun (uint32_t nWifis, uint32_t nSinks, double totalTime, std::string rate,
  std::string
  phyMode, uint32_t nodeSpeed, uint32_t periodicUpdateInterval, uint32_t settlingTime, double dataStart,
  bool
  printRoutes, std::string CSVfileName)
{

```

```

m_nWifis = nWifis; m_nSinks = nSinks; m_totalTime = totalTime; m_rate = rate; m_phyMode =
phyMode;

m_nodeSpeed = nodeSpeed; m_periodicUpdateInterval = periodicUpdateInterval; m_settlingTime =
settlingTime;
m_dataStart = dataStart; m_printRoutes = printRoutes; m_CSVfileName = CSVfileName; std::stringstream
ss;
ss << m_nWifis;
std::string t_nodes = ss.str ();

std::stringstream ss3; ss3 << m_totalTime;
std::string sTotalTime = ss3.str ();

std::string tr_name = "Dsdv_Manet_" + t_nodes + "Nodes_" + sTotalTime + "SimTime"; std::cout
<<"Trace file generated is "<< tr_name <<".tr\n";

CreateNodes (); CreateDevices (tr_name); SetupMobility ();

InstallInternetStack (tr_name); InstallApplications ();

std::cout <<"\nStarting simulation for "<< m_totalTime <<" s ...\n";

CheckThroughput ();

Simulator::Stop (Seconds (m_totalTime)); Simulator::Run ();
Simulator::Destroy ();
}

void DsdvManetExample::CreateNodes ()
{
std::cout <<"Creating "<< (unsigned) m_nWifis <<" nodes.\n"; nodes.Create (m_nWifis);

NS_ASSERT_MSG (m_nWifis > m_nSinks, "Sinks must be less or equal to the number of nodes in
network");
}

void DsdvManetExample::SetupMobility ()
{
MobilityHelper mobility;
ObjectFactory pos;
pos.SetTypeId ("ns3::RandomRectanglePositionAllocator");
pos.Set ("X", StringValue ("ns3::UniformRandomVariable[Min=0.0|Max=1000.0]")); pos.Set ("Y",
StringValue ("ns3::UniformRandomVariable[Min=0.0|Max=1000.0]"));

std::ostringstream speedConstantRandomVariableStream; speedConstantRandomVariableStream
<<"ns3::ConstantRandomVariable[Constant="
<< m_nodeSpeed
<<"]";

```

```

Ptr <PositionAllocator> taPositionAlloc = pos.Create ()->GetObject<PositionAllocator> ();
mobility.SetMobilityModel ("ns3::RandomWaypointMobilityModel", "Speed", StringValue
(speedConstantRandomVariableStream.str ()),
"Pause", StringValue ("ns3::ConstantRandomVariable[Constant=2.0]"), "PositionAllocator",

```

PointerValue

```

(taPositionAlloc)); mobility.SetPositionAllocator (taPositionAlloc); mobility.Install (nodes);
}

```

void

```

DsdvManetExample::CreateDevices (std::string tr_name)
{
NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default (); wifiMac.SetType
("ns3::AdhocWifiMac");
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
YansWifiChannelHelper wifiChannel;
wifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");
wifiChannel.AddPropagationLoss ("ns3::FriisPropagationLossModel"); wifiPhy.SetChannel
(wifiChannel.Create ());
WifiHelper wifi;
wifi.SetStandard (WIFI_PHY_STANDARD_80211b);
wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager", "DataMode", StringValue
(m_phyMode), "ControlMode",
StringValue (m_phyMode));
devices = wifi.Install (wifiPhy, wifiMac, nodes);

```

AsciiTraceHelper ascii;

```

wifiPhy.EnableAsciiAll (ascii.CreateFileStream (tr_name + ".tr")); wifiPhy.EnablePcapAll (tr_name);
}

```

void

```

DsdvManetExample::InstallInternetStack (std::string tr_name)
{
DsdvHelper dsdv;
dsdv.Set ("PeriodicUpdateInterval", TimeValue (Seconds (m_periodicUpdateInterval))); dsdv.Set
("SettlingTime", TimeValue (Seconds (m_settlingTime))); InternetStackHelper stack;
stack.SetRoutingHelper (dsdv); // has effect on the next Install () stack.Install (nodes);
Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");
interfaces = address.Assign (devices); if (m_printRoutes)
{
Ptr<OutputStreamWrapper> routingStream = Create<OutputStreamWrapper> ((tr_name + ".routes"),
std::ios::out);
dsdv.PrintRoutingTableAllAt (Seconds (m_periodicUpdateInterval), routingStream);
}
}

```

```

void
DsdvManetExample::InstallAppli Ptr<Node> node = NodeList::GetNode (i); Ipv4Address nodeAddress =
node->GetObject<Ipv4> ()->GetAddress (1, 0).GetLocal (); Ptr<Socket> sink = SetupPacketReceive
(nodeAddress, node);
}

for (uint32_t clientNode = 0; clientNode <= m_nWifis - 1; clientNode++ )
{
for (uint32_t j = 0; j <= m_nSinks - 1; j++ )
{
OnOffHelper onoff1 ("ns3::UdpSocketFactory", Address (InetSocketAddress (interfaces.GetAddress
(j),
port)));
onoff1.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable[Constant=1.0]"));
onoff1.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable[Constant=0.0]"));

if (j != clientNode)
{
ApplicationContainer apps1 = onoff1.Install (nodes.Get (clientNode)); Ptr<UniformRandomVariable>
var = CreateObject<UniformRandomVariable> (); apps1.Start (Seconds (var->GetValue (m_dataStart,
m_dataStart + 1)));
apps1.Stop (Seconds (m_totalTime));
}
}
}
}

ns3::YansWifiChannelHelper::AddPropagationLoss
void AddPropagationLoss(std::string name, std::string n0="", const AttributeValue
&v0=EmptyAttributeValue(),
std::string n1="", const AttributeValue &v1=EmptyAttributeValue(), std::string n2="", const
AttributeValue
&v2=EmptyAttributeValue(), std::string n3="", const AttributeValue &v3=EmptyAttributeValue(),
std::string n4="",
const AttributeValue &v4=EmptyAttributeValue(), std::string n5="", const AttributeValue
&v5=EmptyAttributeValue(), std::string n6="", const AttributeValue &v6=EmptyAttributeValue(),
std::string n7="",
const AttributeValue &v7=EmptyAttributeValue())

```

Definition: yans-wifi-helper.cc:105 ns3::ApplicationContainer
holds a vector of ns3::Application pointers.

Definition: application-container.h:41 ns3::AsciiTraceHelper
Manage ASCII trace files for device models.

Definition: trace-helper.h:109 ns3::InetSocketAddress
an Inet address class

Definition: inet-socket-address.h:40 ns3::Ptr< Socket > DsdvManetExample

Definition: dsdv-manet.cc:48 ns3::Ipv4InterfaceContainer

holds a vector of std::pair of Ptr<Ipv4> and interface index.

Definition: ipv4-interface-container.h:32 ns3::YansWifiChannelHelper::Create Ptr<YansWifiChannel> Create(void) const

Definition: yans-wifi-helper.cc:153 ns3::WifiHelper::SetRemoteStationManager

```
void SetRemoteStationManager(std::string type, std::string n0="", const AttributeValue
&v0=EmptyAttributeValue(), std::string n1="", const AttributeValue &v1=EmptyAttributeValue(),
std::string n2="",
const AttributeValue &v2=EmptyAttributeValue(), std::string n3="", const AttributeValue
&v3=EmptyAttributeValue(), std::string n4="", const AttributeValue &v4=EmptyAttributeValue(),
std::string n5="",
const AttributeValue &v5=EmptyAttributeValue(), std::string n6="", const AttributeValue
&v6=EmptyAttributeValue(), std::string n7="", const AttributeValue &v7=EmptyAttributeValue())
```

Definition: wifi-helper.cc:68 ns3::StringValue

hold variables of type string

Definition: string.h:19 ns3::YansWifiPhyHelper

Make it easy to create and manage PHY objects for the yans model.

Definition: yans-wifi-helper.h:158 ns3::NqosWifiMacHelper::SetType

```
void SetType(std::string type, std::string n0="", const AttributeValue &v0=EmptyAttributeValue(),
std::string n1="",
const AttributeValue &v1=EmptyAttributeValue(), std::string n2="", const AttributeValue
&v2=EmptyAttributeValue(), std::string n3="", const AttributeValue &v3=EmptyAttributeValue(),
std::string n4="",
const AttributeValue &v4=EmptyAttributeValue(), std::string n5="", const AttributeValue
&v5=EmptyAttributeValue(), std::string n6="", const AttributeValue &v6=EmptyAttributeValue(),
std::string n7="",
const AttributeValue &v7=EmptyAttributeValue())
```

Definition: nqos-wifi-mac-helper.cc:50 NS_LOG_COMPONENT_DEFINE

#define NS_LOG_COMPONENT_DEFINE(name)

Definition: log.h:122 ns3::ObjectFactory::SetTypeId void SetTypeId(TypeId tid)

Definition: object-factory.cc:40 ns3::InternetStackHelper

aggregate IP/TCP/UDP functionality to existing Nodes.

Definition: internet-stack-helper.h:66 ns3::Packet::GetSize

uint32_t GetSize(void) const

Definition: packet.h:620 ns3::AsciiTraceHelper::CreateFileStream

Ptr< OutputStreamWrapper > CreateFileStream(std::string filename, std::ios::openmode
filemode=std::ios::out)

Create and initialize an output stream object we'll use to write the traced bits. ...

Definition: trace-helper.cc:184 ns3::WifiHelper
helps to create WifiNetDevice objects

Definition: wifi-helper.h:92 ns3::OnOffHelper
A helper to make it easier to instantiate an ns3::OnOffApplication on a set of nodes.

Definition: on-off-helper.h:41 ns3::WifiHelper::Install
NetDeviceContainer Install(const WifiPhyHelper &phy, const WifiMacHelper &mac, NodeContainer c)
const

Definition: wifi-helper.cc:97 ns3::Address
a polymorphic address class

Definition: address.h:86 ns3::YansWifiPhyHelper::SetChannel
void SetChannel(Ptr< YansWifiChannel > channel)

Definition: yans-wifi-helper.cc:197 ns3::MobilityHelper::Install
void Install(Ptr< Node > node) const
"Layout" a single node according to the current position allocator type.

Definition: mobility-helper.cc:130 ns3::PcapHelperForDevice::EnablePcapAll
void EnablePcapAll(std::string prefix, bool promiscuous=false)
Enable pcap output on each device (which is of the appropriate type) in the set of all nodes created ...

Definition: trace-helper.cc:432 ns3::TimeValue
hold objects of type ns3::Time

Definition: nstime.h:700 ns3::ObjectFactory::Create Ptr< Object > Create(void) const

Definition: object-factory.cc:89 ns3::NetDeviceContainer
holds a vector of ns3::NetDevice pointers

Definition: net-device-container.h:41 ns3::WifiHelper::SetStandard
void SetStandard(enum WifiPhyStandard standard)

Definition: wifi-helper.cc:91 ns3::MakeCallback
Callback< R > MakeCallback(R(T::*memPtr)(void), OBJ objPtr)

Definition: callback.h:502 ns3::Socket::SetRecvCallback
void SetRecvCallback(Callback< void, Ptr< Socket >>) Notify application when new data is available to be read.

Definition: socket.cc:127 ns3::ApplicationContainer::Start void Start(Time start)
Arrange for all of the Applications in this container to Start() at the Time given as a parameter...

Definition: application-container.cc:84 ns3::NqosWifiMacHelper
create non QoS-enabled MAC layers for a ns3::WifiNetDevice.

Definition: `ngos-wifi-mac-helper.h:33 ns3::CommandLine`

parse command-line arguments
Instances of this class can be used to parse command-line arguments: user...

Definition: `command-line.h:50 ns3::Ipv4`

Access to the Ipv4 forwarding table, interfaces, and configuration.

Definition: `ipv4.h:75 ns3::Config::SetDefault`

`void SetDefault(std::string name, const AttributeValue &value)`

Definition: `config.cc:667 ns3::Socket::Bind`

`virtual int Bind(const Address &address)=0` Allocate a local endpoint for this socket. **ns3::NodeContainer** keep track of a set of node pointers.

Definition: `node-container.h:38 ns3::PointerValue`

hold objects of type `Ptr<T>`

Definition: `pointer.h:33 ns3::Ipv4RoutingHelper::PrintRoutingTableAllAt`

`void PrintRoutingTableAllAt(Time printTime, Ptr< OutputStreamWrapper > stream)` const prints the routing tables of all nodes at a particular time.

Definition: `ipv4-routing-helper.cc:33 ns3::Socket::Recv`

`virtual Ptr< Packet > Recv(uint32_t maxSize, uint32_t flags)=0` Read data from the socket.

ns3::WIFI_PHY_STANDARD_80211b

Definition: `wifi-phy-standard.h:35 ns3::MobilityHelper::SetMobilityModel`

`void SetMobilityModel(std::string type, std::string n1="", const AttributeValue &v1=EmptyAttributeValue(), std::string n2="", const AttributeValue &v2=EmptyAttributeValue(), std::string n3="", const AttributeValue &v3=EmptyAttributeValue(), std::string n4="", const AttributeValue &v4=EmptyAttributeValue(), std::string n5="", const AttributeValue &v5=EmptyAttributeValue(), std::string n6="", const AttributeValue &v6=EmptyAttributeValue(), std::string n7="", const AttributeValue &v7=EmptyAttributeValue(), std::string n8="", const AttributeValue &v8=EmptyAttributeValue(), std::string n9="", const AttributeValue &v9=EmptyAttributeValue())`

Definition: `mobility-helper.cc:79 ns3::UniformRandomVariable::GetValue` double `GetValue(double min, double max)`

Returns a random double from the uniform distribution with the specified range.

Definition: `random-variable-stream.cc:172 ns3::InternetStackHelper::Install`

`void Install(std::string nodeName)` const

Definition: `internet-stack-helper.cc:455 NS_LOG_UNCOND`

`#define NS_LOG_UNCOND(msg)`

Definition: `log.h:343 ns3::ObjectFactory::Set`

`void Set(std::string name, const AttributeValue &value)`

Definition: object-factory.cc:58 ns3::YansWifiChannelHelper
manage and create wifi channel objects for the yans model.

Definition: yans-wifi-helper.h:37 ns3::RngSeedManager::SetSeed static void SetSeed(uint32_t seed)
set the seed it will duplicate the seed value 6 times

Definition: rng-seed-manager.cc:31 NS_ASSERT_MSG
#define NS_ASSERT_MSG(condition, message)

Definition: assert.h:86 ns3::MobilityHelper
Helper class used to assign positions and mobility models to nodes.

Definition: mobility-helper.h:42 ns3::ObjectFactory
instantiate subclasses of ns3::Object.

Definition: object-factory.h:39 ns3::DsdvHelper::Set
void Set(std::string name, const AttributeValue &value)

Definition: dsdv-helper.cc:62 ns3::Ipv4Address
Ipv4 addresses are stored in host order in this class.

Definition: ipv4-address.h:38 ns3::ApplicationContainer::Stop

void Stop(Time stop)
Arrange for all of the Applications in this container to Stop() at the Time given as a parameter...

Definition: application-container.cc:93 ns3::Ipv4AddressHelper::Assign
Ipv4InterfaceContainer Assign(const NetDeviceContainer &c)
Assign IP addresses to the net devices specified in the container based on the current network prefix...

Definition: ipv4-address-helper.cc:131 ns3::CommandLine::AddValue
void AddValue(const std::string &name, const std::string &help, T &value)

Definition: command-line.h:134 ns3::NodeContainer::Get
Ptr< Node > Get(uint32_t i) const
Get the Ptr<Node> stored in this container at a given index.

Definition: node-container.cc:88 ns3::CommandLine::Parse
void Parse(int argc, char *argv[]) const

Definition: command-line.cc:84 ns3::Ipv4AddressHelper
A helper class to make life easier while doing simple IPv4 address assignment in scripts.

Definition: ipv4-address-helper.h:45 ns3::AsciiTraceHelperForDevice::EnableAsciiAll void
EnableAsciiAll(std::string prefix)

Enable ascii trace output on each device (which is of the appropriate type) in the set of all nodes c...

Definition: trace-helper.cc:579 ns3::NodeContainer::Create void Create(uint32_t n)
Create n nodes and append pointers to them to the end of this NodeContainer.

Definition: node-container.cc:93 ns3::YansWifiChannelHelper::SetPropagationDelay
void SetPropagationDelay(std::string name, std::string n0="", const AttributeValue
&v0=EmptyAttributeValue(),
std::string n1="", const AttributeValue &v1=EmptyAttributeValue(), std::string n2="", const
AttributeValue &v2=EmptyAttributeValue(), std::string n3="", const AttributeValue
&v3=EmptyAttributeValue(), std::string n4="",
const AttributeValue &v4=EmptyAttributeValue(), std::string n5="", const AttributeValue
&v5=EmptyAttributeValue(), std::string n6="", const AttributeValue &v6=EmptyAttributeValue(),
std::string n7="",
const AttributeValue &v7=EmptyAttributeValue())

Definition: yans-wifi-helper.cc:129 ns3::MobilityHelper::SetPositionAllocator
void SetPositionAllocator(Ptr< PositionAllocator > allocator)

Definition: mobility-helper.cc:47 ns3::DsdvHelper
Helper class that adds DSDV routing to nodes

Definition: dsdv-helper.h:45 ns3::Object::GetObject
Ptr< T > GetObject(void) const

Definition: object.h:332 ns3::TypeId
a unique identifier for an interface.

Definition: type-id.h:44 ns3::InternetStackHelper::SetRoutingHelper
void SetRoutingHelper(const Ipv4RoutingHelper &routing)

Definition: internet-stack-helper.cc:298 ns3::OnOffHelper::SetAttribute
void SetAttribute(std::string name, const AttributeValue &value)

Definition: on-off-helper.cc:40 ns3::Ipv4AddressHelper::SetBase
void SetBase(Ipv4Address network, Ipv4Mask mask, Ipv4Address base="0.0.0.1") Set the base network
number,
network mask and base address.

Definition: ipv4-address-helper.cc:60 ns3::Ipv4InterfaceContainer::GetAddress Ipv4Address
GetAddress(uint32_t i, uint32_t j=0) const

Definition: ipv4-interface-container.cc:39 ns3::PositionAllocator
Allocate a set of positions. The allocation strategy is implemented in subclasses.

EXPERIMENT NO. 13

Aim: Write a Program in NS3 to Simulate AODV

Source Code:

```
#include "ns3/aodv-module.h" #include "ns3/core-module.h" #include "ns3/network-module.h"
#include "ns3/internet-module.h" #include "ns3/mobility-module.h"
#include "ns3/point-to-point-module.h" #include "ns3/wifi-module.h"
#include "ns3/v4ping-helper.h" #include <iostream>
#include <cmath>

using namespace ns3; class AodvExample
{
public:
    AodvExample ();
    bool Configure (int argc, char **argv); void Run ();
    void Report (std::ostream & os);

private:
    // parameters uint32_t size; doublestep; doubletotalTime; boolpcap; boolprintRoutes;

    // network NodeContainernodes; NetDeviceContainerdevices;
    Ipv4InterfaceContainerinterfaces;

private:
    void CreateNodes (); void CreateDevices ();
    void InstallInternetStack (); void InstallApplications ();
};

int main (int argc, char **argv)
{
    AodvExampletest;
    if (!test.Configure (argc, argv))
        NS_FATAL_ERROR ("Configuration failed. Aborted.");

    test.Run (); test.Report (std::cout); return 0;
}

//-----
AodvExample::AodvExample () : size (10),
step (100),
totalTime (10), pcap (true), printRoutes (true)
{
}
```

```

bool
AodvExample::Configure (int argc, char **argv)
{
// Enable AODV logs by default. Comment this if too noisy
// LogComponentEnable("AodvRoutingProtocol", LOG_LEVEL_ALL);

SeedManager::SetSeed (12345);
CommandLinecmd;

cmd.AddValue ("pcap", "Write PCAP traces.", pcap);

cmd.AddValue ("printRoutes", "Print routing table dumps.", printRoutes); cmd.AddValue ("size",
"Number of nodes.", size);
cmd.AddValue ("time", "Simulation time, s.", totalTime); cmd.AddValue ("step", "Grid step, m", step);

cmd.Parse (argc, argv); return true;
}

void
AodvExample::Run ()
{
// Config::SetDefault ("ns3::WifiRemoteStationManager::RtsCtsThreshold", UIntegerValue (1)); // enable
rts cts all
the time.
CreateNodes (); CreateDevices (); InstallInternetStack (); InstallApplications ();

std::cout <<"Starting simulation for "<<totalTime<<" s ...\n";

Simulator::Stop (Seconds (totalTime)); Simulator::Run ();
Simulator::Destroy ();
}

void
AodvExample::Report (std::ostream &)
{
}

void
AodvExample::CreateNodes ()
{
std::cout <<"Creating "<< (unsigned)size<<" nodes "<<step<<" m apart.\n";
nodes.Create (size);
// Name nodes
for (uint32_t i = 0; i <size; ++i)
{
std::ostringstream os; os <<"node-"<< i;
Names::Add (os.str (), nodes.Get (i));
}
}

```

```

// Create static grid
MobilityHelperpermobility;
mobility.SetPositionAllocator ("ns3::GridPositionAllocator", "MinX", DoubleValue (0.0),
"MinY", DoubleValue (0.0), "DeltaX", DoubleValue (step), "DeltaY", DoubleValue (0), "GridWidth",
UIntegerValue (size),
"LayoutType", StringValue ("RowFirst")); mobility.SetMobilityModel
("ns3::ConstantPositionMobilityModel"); mobility.Install (nodes);
}

v {
Void AodvExample::CreateDevices ()

WifiMacHelper wifiMac; wifiMac.SetType ("ns3::AdhocWifiMac");
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default (); YansWifiChannelHelper wifiChannel =
YansWifiChannelHelper::Default (); wifiPhy.SetChannel (wifiChannel.Create ());
WifiHelperwifi;
wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager", "DataMode", StringValue
("OfdmRate6Mbps"), "RtsCtsThreshold", UintegerValue (0));
devices = wifi.Install (wifiPhy, wifiMac, nodes);

if (pcap)
{
wifiPhy.EnablePcapAll (std::string ("aodv"));
}
}

void
AodvExample::InstallInternetStack ()
{
AodvHelper aodv;
// you can configure AODV attributes here using aodv.Set(name, value)
InternetStackHelperstack;
stack.SetRoutingHelper (aodv); // has effect on the next Install () stack.Install (nodes);
Ipv4AddressHelperaddress; address.SetBase ("10.0.0.0", "255.0.0.0"); interfaces = address.Assign
(devices);

if (printRoutes)
{
Ptr<OutputStreamWrapper> routingStream = Create<OutputStreamWrapper> ("aodv.routes",
std::ios::out); aodv.PrintRoutingTableAllAt (Seconds (8), routingStream);
}
}

void
AodvExample::InstallApplications ()
{
V4PingHelper ping (interfaces.GetAddress (size - 1)); ping.SetAttribute ("Verbose", BooleanValue
(true));

```

```
ApplicationContainer p = ping.Install (nodes.Get (0)); p.Start (Seconds (0));
```

```
p.Stop (Seconds (totalTime) - Seconds (0.001));
```

```
// move node away
```

```
Ptr<Node> node = nodes.Get (size/2);
```

```
Ptr<MobilityModel> mob = node->GetObject<MobilityModel> ();
```

```
Simulator::Schedule (Seconds (totalTime/3), &MobilityModel::SetPosition, mob, Vector (1e5, 1e5, 1e5));
```

```
}
```

```
[2262/2614] Compiling scratch/scratch-simulator.cc
[2530/2614] Linking build/scratch/second
[2531/2614] Linking build/scratch/first
[2532/2614] Linking build/scratch/star
[2568/2614] Linking build/scratch/OSLR-p2p
[2599/2614] Linking build/scratch/static-routing-slash32
[2600/2614] Linking build/scratch/subdir/subdir
[2601/2614] Linking build/scratch/AODV
[2602/2614] Linking build/scratch/scratch-simulator
[2603/2614] Linking build/scratch/DSDV
Waf: Leaving directory `/home/mait/Downloads/ns-allinone-3.26/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (16.495s)
Creating 10 nodes 100 m apart.
Starting simulation for 10 s ...
PING 10.0.0.10 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=0 ttl=56 time=2057 ms
64 bytes from 10.0.0.10: icmp_seq=1 ttl=56 time=1059 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=56 time=60 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=56 time=8 ms
--- 10.0.0.10 ping statistics ---
10 packets transmitted, 4 received, 60% packet loss, time 9999ms
rtt min/avg/max/mdev = 8/796/2057/969.9 ms
[mait@CSE-114B-6 ns-3.26]$
```

EXPERIMENT NO. 14

Aim : Write a Program in NS3 to Simulate OLSR

Source Code:

OLSR routing over some point-to-point links

Network topology

// n0

\ 5 Mb/s, 2ms

\ 1.5Mb/s, 10ms

n2 -----n3-----n4

/

/ 5

Mb/s,

2ms

n1

// - all links are point-to-point links with indicated one-way BW/delay

// - CBR/UDP flows from n0 to n4, and from n3 to n1

// - UDP packet size of 210 bytes, with per-packet interval 0.00375 sec.

// (i.e., DataRate of 448,000 bps)

// - DropTail queues

// - Tracing of queues and packet receptions to file "simple-point-to-point-olsr.tr"

#include <iostream> #include <fstream> #include <string> #include <cassert>

#include "ns3/core-dule.h" #include "ns3/network-module.h" #include "ns3/internet-module.h"

#include "ns3/point-to-point-module.h" #include "ns3/applications-module.h" #include "ns3/olsr-helper.h"

#include "ns3/ipv4-static-routing-helper.h"


```

#include "ns3/ipv4-list-routing-helper.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("SimplePointToPointOlsrExample");

int
main (int argc, char *argv[])
{
// Users may find it convenient to turn on explicit debugging
// for selected modules; the below lines suggest how to do this #if 0
LogComponentEnable ("SimpleGlobalRoutingExample", LOG_LEVEL_INFO); #endif

// Set up some default values for the simulation. Use the

Config::SetDefault ("ns3::OnOffApplication::PacketSize", UIntegerValue (210));
Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue ("448kb/s"));

//DefaultValue::Bind ("DropTailQueue::m_maxPackets", 30);

// Allow the user to override any of the defaults and the above
// DefaultValue::Bind ()s at run-time, via command-line arguments
CommandLineCmd; cmd.Parse (argc, argv);

// Here, we will explicitly create four nodes. In more sophisticated
// topologies, we could configure a node factory. NS_LOG_INFO ("Create nodes."); NodeContainer c;
Create (5);
NodeContainer n02 = NodeContainer (c.Get (0), c.Get (2)); NodeContainer n12 = NodeContainer
(c.Get (1), c.Get (2)); NodeContainer n32 = NodeContainer (c.Get (3), c.Get (2)); NodeContainer n34
= NodeContainer (c.Get (3), c.Get (4));

// Enable OLSR
NS_LOG_INFO ("Enabling OLSR Routing.");
OlsrHelperolsr;

Ipv4StaticRoutingHelper staticRouting;

Ipv4ListRoutingHelperlist; list.Add (staticRouting, 0);
list.Add (olsr, 10);

InternetStackHelper internet;
internet.SetRoutingHelper (list); // has effect on the next Install () internet.Install (c);

// We create the channels first without any IP addressing information
NS_LOG_INFO ("Create channels.");
PointToPointHelper p2p;

```

```

p2p.SetDeviceAttribute ("DataRate", StringValue ("5Mbps")); p2p.SetChannelAttribute ("Delay",
StringValue ("2ms")); NetDeviceContainer nd02 = p2p.Install (n02); NetDeviceContainer nd12 =
p2p.Install (n12); p2p.SetDeviceAttribute ("DataRate", StringValue ("1500kbps"));

p2p.SetChannelAttribute ("Delay", StringValue ("10ms")); NetDeviceContainer nd32 = p2p.Install
(n32); NetDeviceContainer nd34 = p2p.Install (n34);

// Later, we add IP addresses. NS_LOG_INFO ("Assign IP Addresses."); Ipv4AddressHelper ipv4;
ipv4.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer i02 = ipv4.Assign (nd02);

ipv4.SetBase ("10.1.2.0", "255.255.255.0");
Ipv4InterfaceContainer i12 = ipv4.Assign (nd12);

ipv4.SetBase ("10.1.3.0", "255.255.255.0");
Ipv4InterfaceContainer i32 = ipv4.Assign (nd32);

ipv4.SetBase ("10.1.4.0", "255.255.255.0");
Ipv4InterfaceContainer i34 = ipv4.Assign (nd34);

// Create the OnOff application to send UDP datagrams of size
// 210 bytes at a rate of 448 Kb/s from n0 to n4 NS_LOG_INFO ("Create Applications."); uint16_t port =
9; //
Discard port (RFC 863)

OnOffHelper onoff ("ns3::UdpSocketFactory", InetSocketAddress (i34.GetAddress (1), port));
onoff.SetConstantRate (DataRate ("448kb/s"));

ApplicationContainer apps = onoff.Install (c.Get (0)); apps.Start (Seconds (1.0));
apps.Stop (Seconds (10.0));

// Create a packet sink to receive these packets PacketSinkHelpersink ("ns3::UdpSocketFactory",
InetSocketAddress (Ipv4Address::GetAny (), port));

apps = sink.Install (c.Get (3)); apps.Start (Seconds (1.0));
apps.Stop (Seconds (10.0));

// Create a similar flow from n3 to n1, starting at time 1.1 seconds onoff.SetAttribute ("Remote",
AddressValue (InetSocketAddress (i12.GetAddress (0), port)));
apps = onoff.Install (c.Get (3)); apps.Start (Seconds (1.1));
apps.Stop (Seconds (10.0));

// Create a packet sink to receive these packets apps = sink.Install (c.Get (1));
apps.Start (Seconds (1.1));
apps.Stop (Seconds (10.0));

```

```
AsciiTraceHelper ascii;  
p2p.EnableAsciiAll (ascii.CreateFileStream ("simple-point-to-point-olsr.tr")); p2p.EnablePcapAll  
("simple-point-to-point-olsr");
```

```
Simulator::Stop (Seconds (30));
```

```
NS_LOG_INFO ("Run Simulation."); Simulator::Run (); Simulator::Destroy (); NS_LOG_INFO  
("Done.");
```

```
return 0;  
}
```