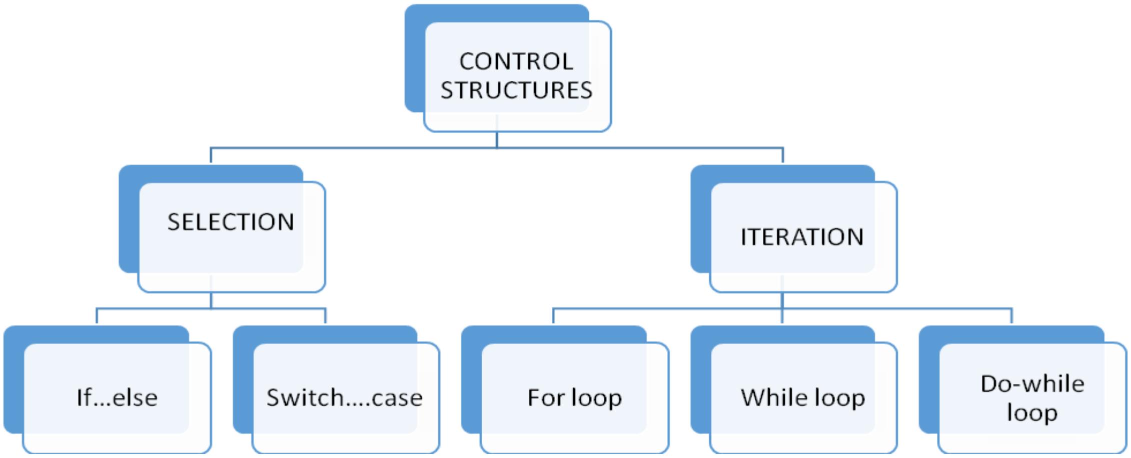
SELECTION STATEMENTS

CONTROL STRUCTURES

- Control structures control the order in which the statements are executed.
- There are two main categories of control structures namely Selection statements and Iteration statements.



SELECTION STATEMENTS

- The selection statement allows us to choose set of instructions for execution depending upon the truth value of the expression.
- There are two types of selection statements: if and switch.
- The selection statements are also called *conditional statements* or *decision statements*.

if...else STATEMENT

Syntax:

if (conditional expression)

```
{
Statement Block;
}
else
{
Statement Block;
}
```

- ❑ if statement allows selection depending upon the outcome of a condition.
- ❑ If the conditional expression evaluates to true, then the statement immediately following if will be executed.
- ❑ If the conditional expression evaluates to false, then the statements following the else clause will be executed.
- ❑ Always enclose the conditional expression in parenthesis.
- ❑ The conditional expression may be a simple or compound. In the case of simple expression there is only one relational expression. In case of compound expression there are more than one relational expression which are joined by the logical operator.
- ❑ Each statement block (if-else) may have a single or multiple statement. In the case of a single statement, it is not mandatory to enclose it in curly braces ({}) but if there are multiple statements then they must be enclosed in curly braces ({}).
- ❑ The else clause is optional and needs to be included only when some action is to be taken if the test condition evaluates to false.
- ❑ Semicolon should not be placed after the parenthesis of the conditional expression of the if statement or else statement.

RELATIONAL OPERATORS

A relational operator is used to test for some kind of relation between two entities.

For eg.

```
int Sum=15
(Sum = =15 )    //true
(Sum !=15)      //false
(Sum >10)       // true
(Sum <10)      //false
(Sum >=15)   //true
```

OPERATOR	PURPOSE
= =	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than equal to
<=	Less than equal to

LOGICAL OPERATORS

Logical operators are used to join two or more relational expressions.

OPERATOR	PURPOSE
!	NOT (Negates the condition)
||	OR (gives TRUE if any of the condition is TRUE)
AND	AND (Gives TRUR if both the conditions are TRUE)

For eg.

```
int Sum=15, num=5;
```

<table>
<tr><td>(Sum = =15) &&
(num>10)</td><td>The condition is false as first condition is true but the second condition is false. The condition can be true if both the conditions are true.</td></tr>
<tr><td>(Sum = =15) ||
(num>10)</td><td>The condition is true as first condition is true but the second condition is false. The condition can be true if any of the conditions are true.</td></tr>
<tr><td>! (Sum ==15)</td><td>The condition is false because sum is 15 and is true and
(! True) is false.</td></tr>
</table>

Click to view video on if else statement

EXAMPLES OF TEST CONDITIONS

Let us have a look at some examples of test conditions

Test conditions	Explanations

if(grade==‘a’)
System.out.println ("You did well ");
	comparing the value stored in variable grade with constant value ‘a’.

if(a>b) System.out.println ("a has more than b has ");	comparing values of two variables

if(x)
{ System.out.println ("x is non-zero ");
System.out.println ("it results into true
	testing truth value of a variable. If the value of the variable x is zero if evaluates to false. If the value of the variable x is non-

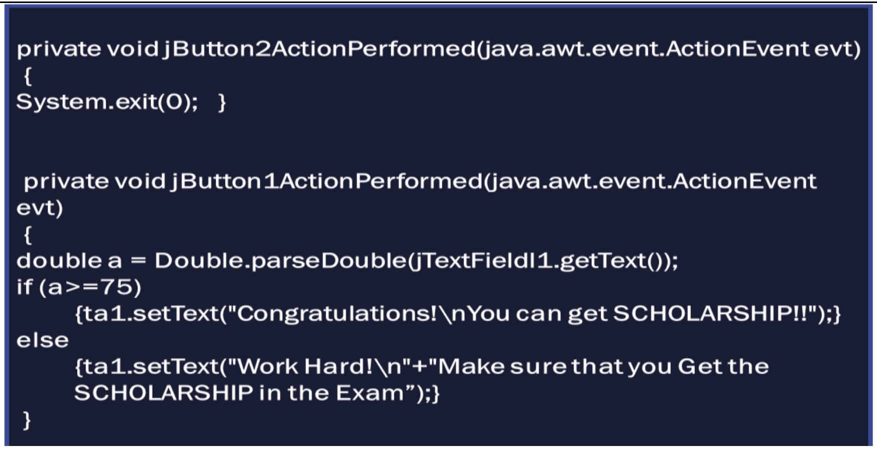
");
}
	zero if evaluates to true.

if ((x%2==0) && (x>100))
System.out.println ("great ");
	Here conditional statement is a compound expression. Two conditions are joined by the logical operator &&. It evaluates to true if both the conditions are true.

int x=5;
if (x=10)
System.out.println ("done ");
else
System.out.println ("not done ");
	Here (x=10) is an assignment expression. It assigns value 10 to x and the value of the expression also becomes 10. In c++ any number other than 0 is considered as true. Thus 10 is considered as true and this expression always evaluates to true irrespective of the value of x.

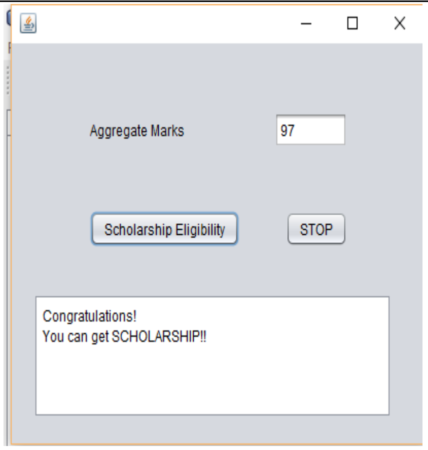
EXAMPLE - if with else

Q WAP to input average marks of student and display the appropriate message.



```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
 {
System.exit(0);   }


 private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt)
 {
double a = Double.parseDouble(jTextFieldl1.getText());
if (a>=75)
      {ta1.setText("Congratulations!\nYou can get SCHOLARSHIP!!");}
else
      {ta1.setText("Work Hard!\n"+"Make sure that you Get the
      SCHOLARSHIP in the Exam");}
 }
```

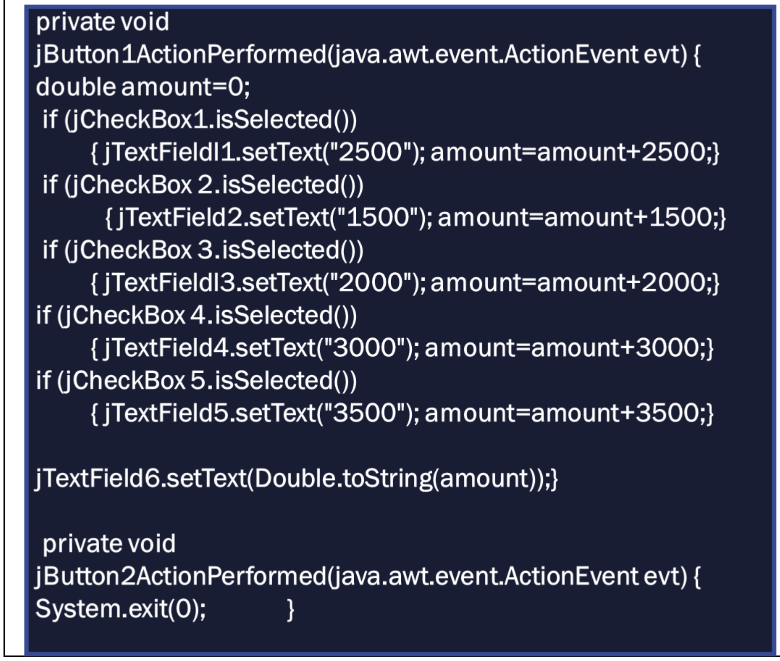


Aggregate Marks: 97

Congratulations!
You can get SCHOLARSHIP!!

EXAMPLE - CHECK BOX

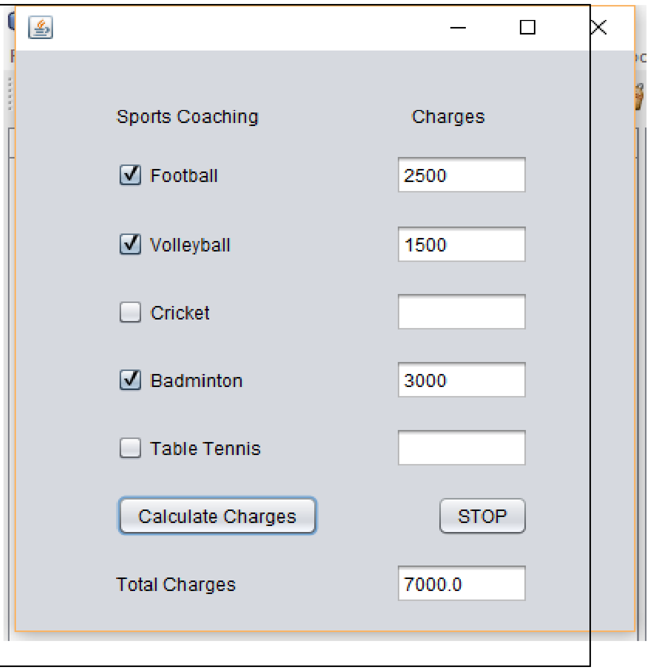
Q WAP to calculate the total charges depending upon the games selected by the user.



```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
double amount=0;
 if (jCheckBox1.isSelected())
        { jTextFieldl1.setText("2500"); amount=amount+2500;}
 if (jCheckBox 2.isSelected())
          { jTextField2.setText("1500"); amount=amount+1500;}
 if (jCheckBox 3.isSelected())
        { jTextFieldl3.setText("2000"); amount=amount+2000;}
if (jCheckBox 4.isSelected())
        { jTextField4.setText("3000"); amount=amount+3000;}
if (jCheckBox 5.isSelected())
        { jTextField5.setText("3500"); amount=amount+3500;}

jTextField6.setText(Double.toString(amount));}

 private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
System.exit(0);          }
```

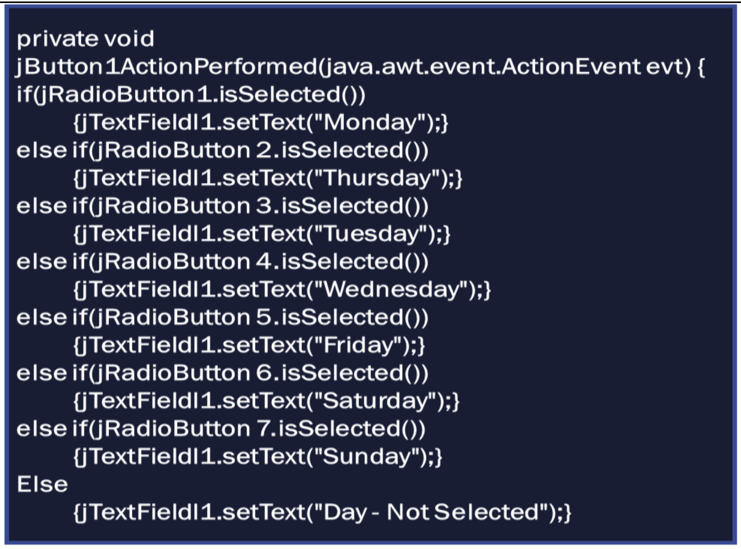


Sports Coaching	Charges
☑ Football	2500
☑ Volleyball	1500
□ Cricket	
☑ Badminton	3000
□ Table Tennis	

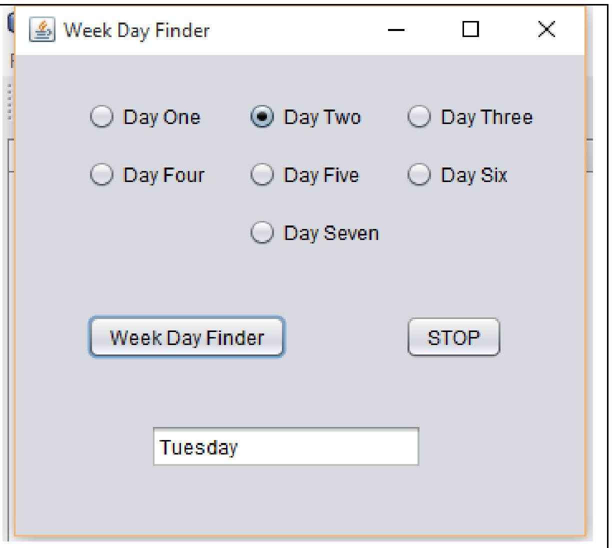
Total Charges 7000.0

EXAMPLE - RADIO BUTTON

Q WAP to select the day number and display the corresponding day name.



```
private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
if(jRadioButton1.isSelected())
        {jTextFieldl1.setText("Monday");}
else if(jRadioButton 2.isSelected())
        {jTextFieldl1.setText("Thursday");}
else if(jRadioButton 3.isSelected())
        {jTextFieldl1.setText("Tuesday");}
else if(jRadioButton 4.isSelected())
        {jTextFieldl1.setText("Wednesday");}
else if(jRadioButton 5.isSelected())
        {jTextFieldl1.setText("Friday");}
else if(jRadioButton 6.isSelected())
        {jTextFieldl1.setText("Saturday");}
else if(jRadioButton 7.isSelected())
        {jTextFieldl1.setText("Sunday");}
Else
        {jTextFieldl1.setText("Day - Not Selected");}
```

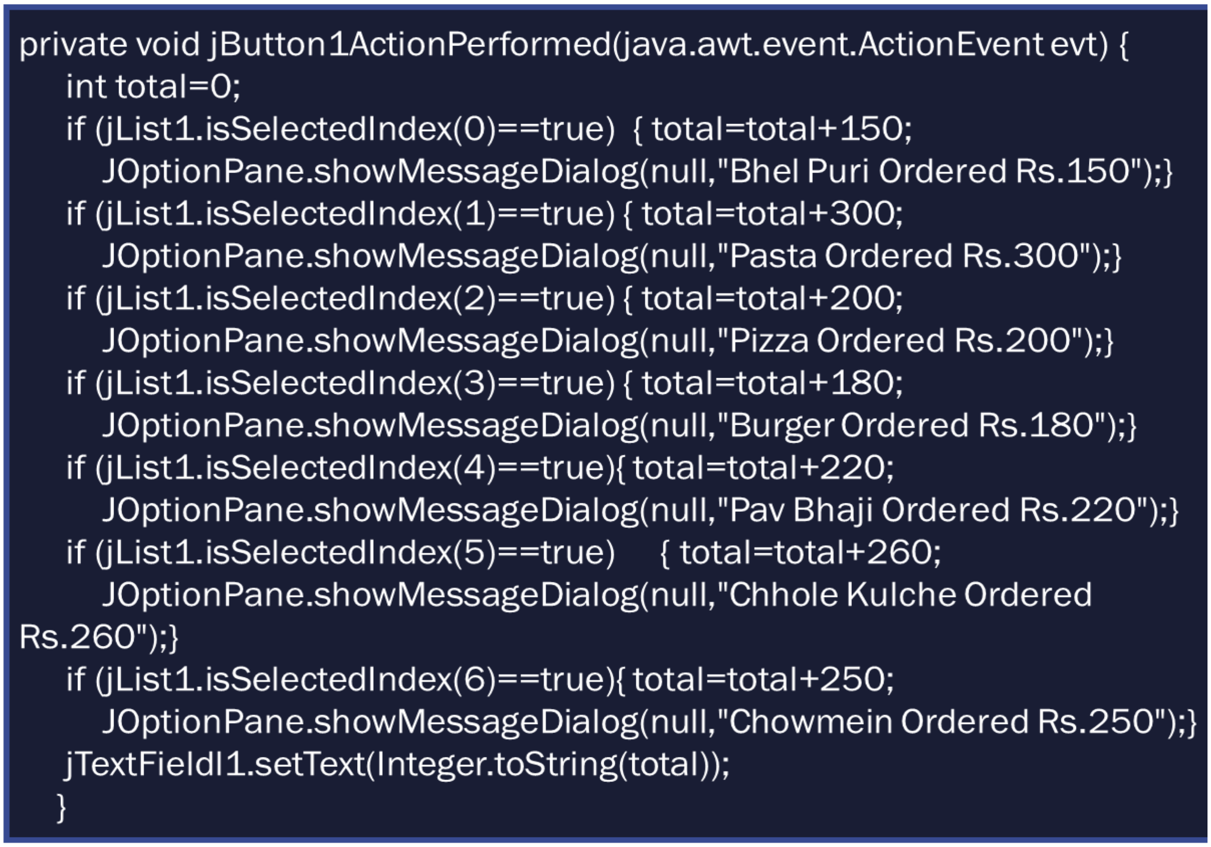


○ Day One ● Day Two ○ Day Three
○ Day Four ○ Day Five ○ Day Six
○ Day Seven

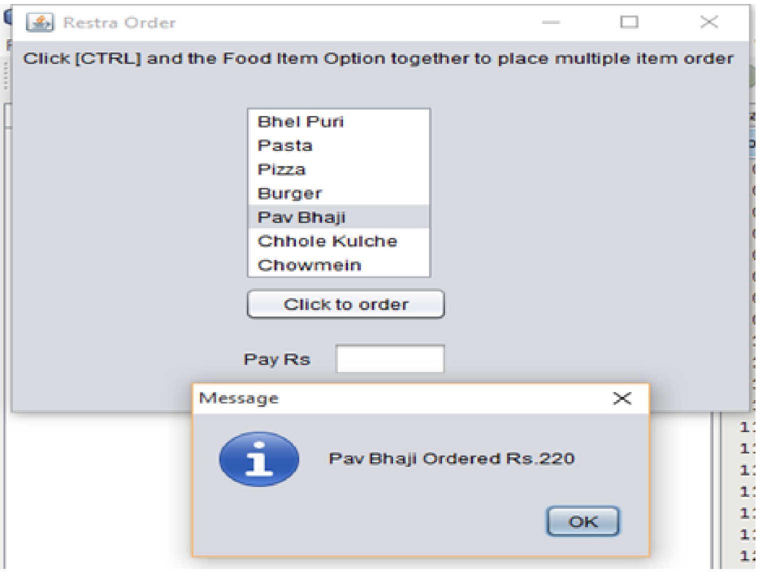
Tuesday

EXAMPLE - LIST

Q WAP to select the food items from the list and calculate the total amount the user has to pay.



```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    int total=0;
    if (jList1.isSelectedIndex(0)==true)  { total=total+150;
       JOptionPane.showMessageDialog(null,"Bhel Puri Ordered Rs.150");}
    if (jList1.isSelectedIndex(1)==true) { total=total+300;
       JOptionPane.showMessageDialog(null,"Pasta Ordered Rs.300");}
    if (jList1.isSelectedIndex(2)==true) { total=total+200;
       JOptionPane.showMessageDialog(null,"Pizza Ordered Rs.200");}
    if (jList1.isSelectedIndex(3)==true) { total=total+180;
       JOptionPane.showMessageDialog(null,"Burger Ordered Rs.180");}
    if (jList1.isSelectedIndex(4)==true){ total=total+220;
       JOptionPane.showMessageDialog(null,"Pav Bhaji Ordered Rs.220");}
    if (jList1.isSelectedIndex(5)==true)     { total=total+260;
       JOptionPane.showMessageDialog(null,"Chhole Kulche Ordered
Rs.260");}
    if (jList1.isSelectedIndex(6)==true){ total=total+250;
       JOptionPane.showMessageDialog(null,"Chowmein Ordered Rs.250");}
    jTextFieldl1.setText(Integer.toString(total));
   }
```



Restra Order

Click [CTRL] and the Food Item Option together to place multiple item order

Bhel Puri
Pasta
Pizza
Burger
Pav Bhaji
Chhole Kulche
Chowmein

Pay Rs

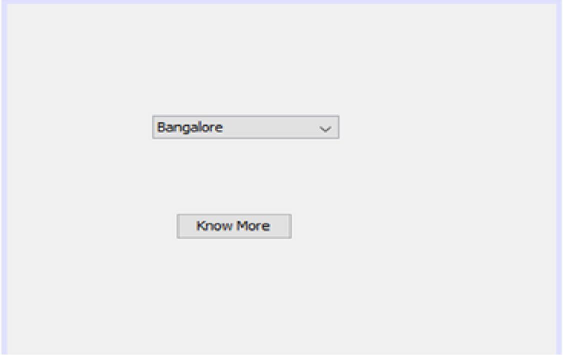
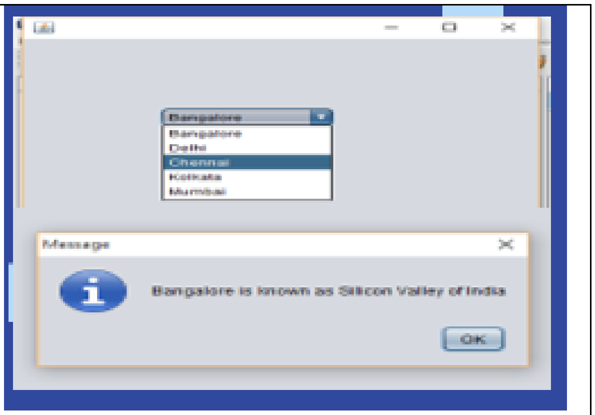
Message

Pav Bhaji Ordered Rs.220

EXAMPLE - COMBO BOX

Q WAP to select city from the combo box and display the appropriate message.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
if(c1.getSelectedIndex() == 0)
      JOptionPane.showMessageDialog(this,c1.getSelectedItem()+" is
      known as Silicon Valley of India");
if(c1.getSelectedIndex() == 1)
      JOptionPane.showMessageDialog(this,c1.getSelectedItem()+" is
      known as Capital City of India");
if(c1.getSelectedIndex() == 2)
      JOptionPane.showMessageDialog(this,c1.getSelectedItem()+" is
      known for broad industrial base of India");
if(c1.getSelectedIndex() == 3)
      JOptionPane.showMessageDialog(this,c1.getSelectedItem()+" is
      known for literary,artistic and revolutionary heritage of India");
if(c1.getSelectedIndex() == 4)
      JOptionPane.showMessageDialog(this,c1.getSelectedItem()+" is
      known for hub in of Bollywood");
else
      JOptionPane.showMessageDialog(this, "No City Selected");
}   }
```

	 Bangalore Bangalore Delhi Chennai Kolkata Mumbai Message OK Bangalore is known as Silicon Valley of India

switch....case STATEMENT

SYNTAX:

```
switch (variable/expression)
{
case constant 1: statement sequence 1;
 break;
case constant 2: statement sequence 2;
break;
    .
    .
case constant n: statement sequence n;
break;
default:  statement invalid;
}
```

❑ Switch is a multiple branch selection statement.

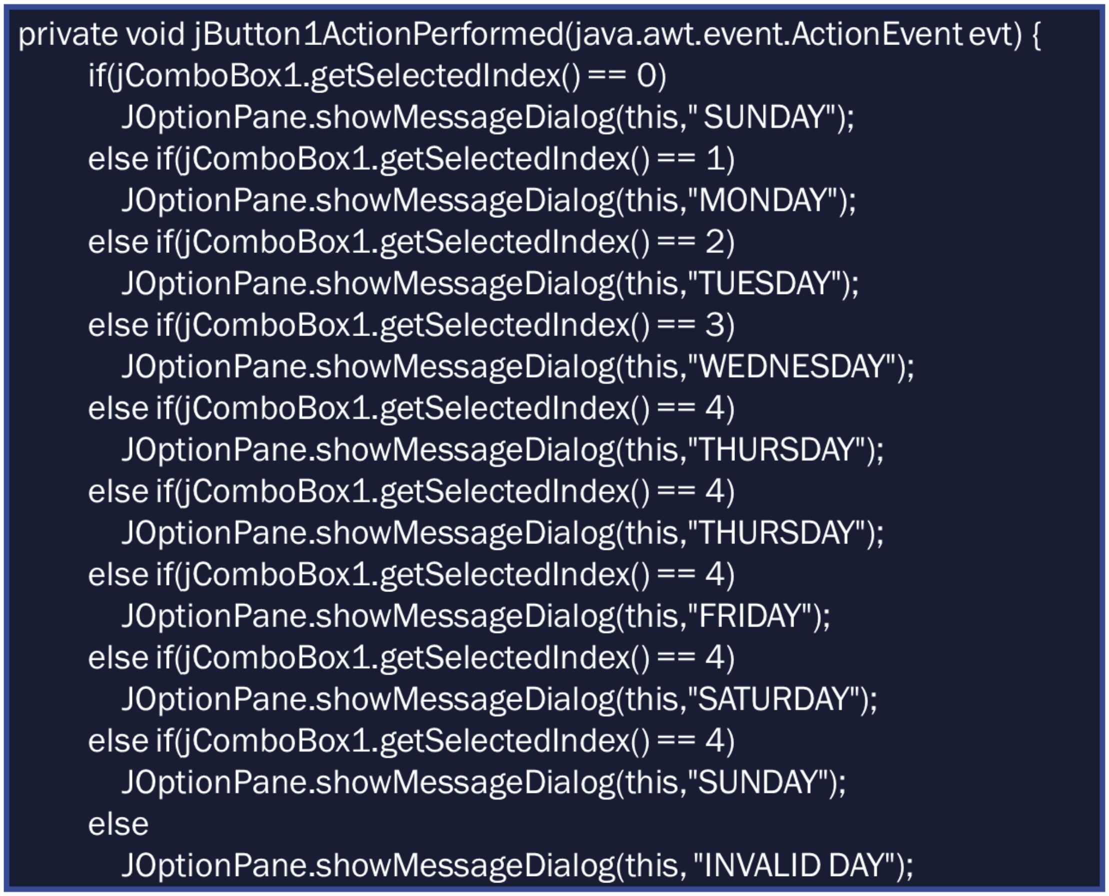
❑ A switch statement allows a variable to be tested for equality against a list of values.

- ❑ Each value is called a case and the variable being switched on is checked for each case. When a match is found, the statements associated with that constant are executed.
- ❑ It can have as many numbers of case statements within a switch. Each case is followed by the value to be compared to and a colon (:).
- ❑ The constant expression for a case must be the same data type as the variable in the switch, and it must be a constant or literal.
- ❑ When the variable being switched on is equal to the case, the statement following the case will execute until a break statement is reached.
- ❑ It has a default case which appears at the end of the switch case. It is used for performing a task when none of the task is true. No break is needed in the default case.
- ❑ The expression in switch statement must either evaluate to an integer value or a character value. It cannot be a String or a real number.
- ❑ Default and break are optional in switch case.

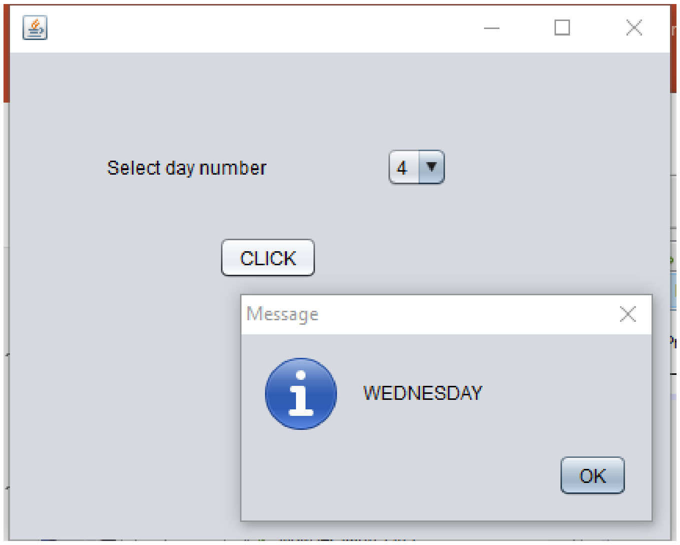
Click to View Video on Nested if else and switch case

EXAMPLE - SWITCH CASE

Q WAP to input a number and display the corresponding week day.



```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jComboBox1.getSelectedIndex() == 0)
      JOptionPane.showMessageDialog(this," SUNDAY");
    else if(jComboBox1.getSelectedIndex() == 1)
      JOptionPane.showMessageDialog(this,"MONDAY");
    else if(jComboBox1.getSelectedIndex() == 2)
      JOptionPane.showMessageDialog(this,"TUESDAY");
    else if(jComboBox1.getSelectedIndex() == 3)
      JOptionPane.showMessageDialog(this,"WEDNESDAY");
    else if(jComboBox1.getSelectedIndex() == 4)
      JOptionPane.showMessageDialog(this,"THURSDAY");
    else if(jComboBox1.getSelectedIndex() == 4)
      JOptionPane.showMessageDialog(this,"THURSDAY");
    else if(jComboBox1.getSelectedIndex() == 4)
      JOptionPane.showMessageDialog(this,"FRIDAY");
    else if(jComboBox1.getSelectedIndex() == 4)
      JOptionPane.showMessageDialog(this,"SATURDAY");
    else if(jComboBox1.getSelectedIndex() == 4)
      JOptionPane.showMessageDialog(this,"SUNDAY");
    else
      JOptionPane.showMessageDialog(this, "INVALID DAY");
```

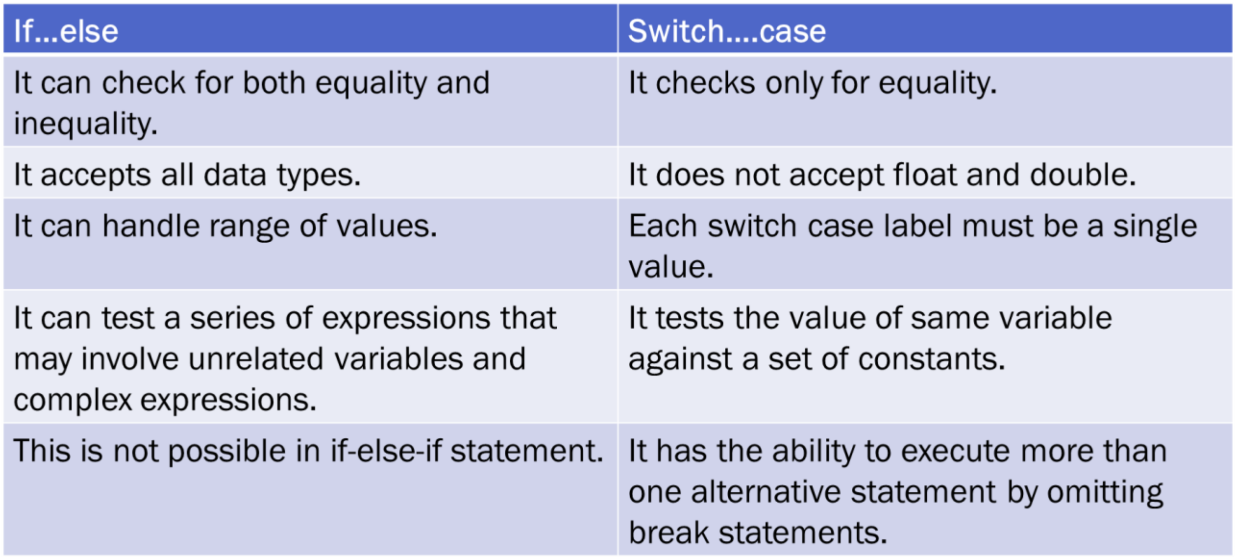


Select day number 4

Message

WEDNESDAY

DIFFERENCE BETWEEN if….else AND switch….case



If…else	Switch….case
It can check for both equality and inequality.	It checks only for equality.
It accepts all data types.	It does not accept float and double.
It can handle range of values.	Each switch case label must be a single value.
It can test a series of expressions that may involve unrelated variables and complex expressions.	It tests the value of same variable against a set of constants.
This is not possible in if-else-if statement.	It has the ability to execute more than one alternative statement by omitting break statements.