

## Chapter 6 : Java IDE Programming – I , II & III

In Java, the GUI programmin is done through Swing API ( Application Programming Interface).

Swing is a set of classes that provides more powerful and flexible components than are possible with AWT (Abstract Windows Toolkit). It supplies buttons, chekboxes, labels, tabbed panes, scrool pans, trees, tables, dialog boxes etc.

The swing controls/components are categorized as:

**o Components :** Self contained graphic entity ( eg. JLabel, JBotten, JtextField etc)

**o Containers:** Component that can hold other components (eg. JPanel, JFrame, JDialog, JWindow)

### COMMONLY AVAILABLE SWING CONTROLS IN JAVA

**o JFrame:** A Frame is a container control, in which all the controls can be lace.

**o JLabel:** JLable allows placing un-editable text on the Frame/Panel

**o jTextField:** JTextFeild allows placing editable text on the Frame/Pane. User can enter text in a text field during runtime.

**o jbutton:** is used to initiate an action when it is clicked.

**o jList:** is a group of values or items from which one or more selections can be made.

**o jComboBox:** jComboBox is similar to jList but also allow to enter editable text during run time. It is a combination of jTextFiled and jList.

**o jRadioButton:** Allow us to choose a single item from a group of jRadioButton options.

**o jCheckBox:** Allow us to choose one or more items from a group of jCheckBox options.

**o jPasswordField:** Allow us to enter a text during the run time but shows an encrypted text instead of the original text

**o jTextArea:** JTextArea is a multi-line text component to enter or edit text.

**o Focus:** The control under execution is said to have the focus. The control having the focus obtains input form the user.

**o getText():** getText() method is used to obtain the text from a jTextFeild during the run time.

**o setText():** setText() method is used to set or change the text of a jTextFeild during run time.

### Swing Controls Methods and Properties:

These are the Swing Controls available with NetBeans IDE and their concern methods and properties are given below.

### Some Important Questions with Answers

1. Which window is used to design the form.

**Ans:** Design window

2. Which window contains the Swing Controls components.

**Ans:** Palette window

3. What is the most suitable component to accept multiline text.

**Ans:** Text Area

**Ans:** Text Area

4. What will be the output of the following command? `Learning.concat("Java")`

**Ans:** Error

5. What will be the output of the following command? `"Learning".concat("Java")`

**Ans:** LearningJava

6. Name the different list type controls offered by Java Swing.

**Ans:**

(i) `JList`

(ii) `JComboBox`

7. Name any two commonly used methods of `List`.

**Ans:** `getSelectedIndex()` and `getSelectedValue()`

8. Write code to add an element ("New Course") to a list (`SubList`) at the beginning of the list.

**Ans:** `SubList.add(0,"New Course");`

9. Describe the three common controls. Also give some of their properties.

10. By default a combo box does not offer editing feature. How would you make a combo box editable.

**Ans:** By setting its editable property to false.

11. Write Name the component classes of Swing API for the following components- a) frame (b) button

**Ans:** (a) JFrame (b) JButton

12. What is the name of event listener interface for action events ?

**Ans:** ActionListener

13. What does getPassword() on a password field return ?

**Ans:** A character array.

14. What is event driven programming?

**Ans:** This programming style responds to the user events and is driven by the occurrence of user events.

15. What are containers? Give examples.

**Ans:** Containers are those controls inside them e.g., frame (JFrame), Panel (JPanel), label (JLabel) etc. are containers.

16. Which method of list is used to determine the value of selected item, if only one item is selected?

**Ans:** getSelectedValue()

17. Which type of storage is provided by variables?

**Ans:** Temporary

18. What will be the output of the following code segment:

```
String firstName = "Manas ";  
String lastName = "Pranav";  
String fullName = firstName + lastName;  
jTextField1.setText("Full Name: ");  
jTextField2.setText (fullName);
```

**Ans:** Full Name: ManasPranav

19. Which expression is used to print the value of a variable "x" of type int.

**Ans:** jTextField1.setText("x = " + x);

20. The statement i++; is equivalent to

**Ans:** i= i+1

21. Name the primitive datatypes in java.

**Ans:** Numeric type , Fractional type, Character type and Boolean type.

22. Which events get fired when a user clicks a JButton and JRadioButton.

**Ans:** ActionPerformed

23. Which of the following is a selection construct?

a. do while Loop b. for Loop c. while Loop d. None of these

**Ans:** d . None of these

24. What will be used if there are two or more possible options?

**Ans:** We can use if.....else conditional statement or switch.....case statement.

25. Name the loop that never ends.

**Ans:** Infinite loop. For example:

```
for( k=1;k<=10;k++)  
{  
System.out.println("It is infinite loop"); k=k- 1;  
}
```

26. Which braces are used to enclose statements in a block statement.

**Ans:** { } Curly braces

27. Which of the following is an exit controlled loop?

a. for loop b. do while Loop c. while loop d. none of these

**Ans:** do.... while loop

28. Which process is used to translate a task into a series of commands that a computer will use to perform that task.

**Ans:** Project design

29. Which of the following component is the best suited to accept the country of the user?

A. List B Combo box C Radio button D Check box

**Ans:** List and combo box

30. Which construct will be used to find the sum of the first 10 natural numbers?

**Ans:** for loop

31. Which of the following is not a good programming guideline?

**Ans:** Using text fields to accept input of marital status

### **SHORT ANSWERS TYPE QUESTIONS**

1. Explain the following terms:

a) IDE b) Form

**Ans:** IDE : IDE is an acronym for Integrated Development Environment which is a work environment that integrates all tools necessary for Application Development and makes them available as part of one environment.

b) Forms: Forms are used to accept data (input) and submit data to an external agent for processing.

2. Explain the usage of the following methods :

a) setText()

b) toString() concat()

**Ans:** a) setText() :

It is used to change the display text of a component (label, text field or button) during run time.

b) toString() : It is used to convert an Integer value to String type

c) concat() : The concat() method or the string concatenation symbol(+) may be used to add two strings together.

3. Differentiate between:

a) Text field and Text area components :

The Text Field allows the user to enter a single line of text only. But Text Area component allows to accept multiline input from the user or display multiple lines of information.

b) Text field and Password field components:

The Text Field displays the obtained text in unencrypted form whereas password field displays the obtained text in encrypted form. This component allows confidential input like passwords which are single line.

c)parseInt() and parseDouble() methods: parseInt() is used to convert a string value to Integer type whereas parseDouble() is used to convert a string value to type Double.

4.What is a Variable?

**Ans:** Variables are named temporary storage locations.

5.Why are data types important?

**Ans:** Data Types define the way the values are stored, the range of the values and the operations that can be performed on that type.

6.How are keywords different from variable names?

**Ans:** Keywords have special meaning in java, should not be used as the variable names. Variables are named temporary storage locations.

7.What is an identifier?

**Ans:**Identifiers are fundamental building block of a program and are used as the general terminology for the names given to different parts of the program viz. variables, objects, classes,functions, arrays etc.

8.What is casting? When do we need it?

**Ans:**Casting is a conversion, which uses the cast operator to specify the type name in parenthesis and is placed in front of the value to be converted.

**For example:** Result = (float) total / count ;

They are helpful in situations where we temporarily need to treat a value as another type.

9.What is the purpose of break statement in a loop?

**Ans:**In a loop, the break statement terminates the loop when it gets executed.

10.Is Java case sensitive? What is meant by case sensitive?

**Ans:** Yes java is case sensitive. Case sensitive means upper case letters and lower case letters are treated differently.

11. Is a string containing a single character same as a char?

**Ans:** No

12.What is the main difference between a combo box and a list box?

**Ans:** The List Box does not have a text field the user can use to edit the selected item, whereas a Combo Box is cross between a text field and a list.

13.Explain the use of for statement along with its syntax.

**Ans:** The for loop repeat a set of statements till a test condition is satisfied. The syntax of the for loop is:

Syntax:

```
for( initialization; test exp; increment/decrement exp)
{
statements;
}
```

14.What is the difference between selection and repetition?

**Ans:** Selection statements test conditions and selectively execute code depending on the outcome of the test condition , whereas repetition statements repeat a set of statements till a test condition is satisfied.

15.What is the purpose of default clause in a switch statement?

**Ans:** The default statement gets executed when no match is found in switch.

16.What is the main difference between a while loop and a do while loop?

**Ans:** In while loop test expression is evaluated at the beginning of the loop whereas in do while loop the test expression is evaluated at the bottom of the loop. This means that do-while loop is executed at least once.

17.How is the if...else if combination more general than a switch statement?

**Ans:**The switch statement must be by a single integer control variable, and each case section must correspond to a single constant value for the variable. The if...else if combination allows any kind of condition after each if.

18.Excessive comments add time to the execution of your program. (True/False).

**Ans:** False because comments are non executable.

19.Differentiate between compile time and run time errors.

**Ans:** Compile time errors refer to the errors that violate the grammatical rules and regulations of programming language.

20.Which error is harder to locate and why?

**Ans:** Logical errors because in presence of logical error , the program executes without any problems but the output produced is not correct. Therefore every statement of the program need to be scanned.

21.Explain the following terms:

**a) Exception Handling:** Run time errors are also called exceptions, and handling such errors in the application is called exception handling.

**b)Syntax:** Formal set of rules defined for writing any statement in a language is known as syntax.

**c)Portability:** The application should be portable. It should be able to run on different platforms.

**d) Prettyprinting:** Prettyprinting is the formatting of a program to make it more readable. These formatting conventions usually consist of changes in positioning, spacing, color, contrast, size and similar modifications intended to make the content easier to view, read and understand.

**e)Syntax error:** Syntax errors occur when syntax rules of any programming language are violated. These errors occur during compilation of the application

22. What event is fired when you click a JButton?

**Ans:** Action Event

23. Write the use of following methods used with check box control.

a) getText( ),

b) setText(String s)

c) isSelected( )

d) setSelected( )

**Ans a)** getText( ) Returns the text displayed by the checkbox

b) setText(String s) Sets the text displayed by the check box to the String value specified in parenthesis.

c) isSelected( ) Returns the state of check box - true if selected else returns false.

d) setSelected( ) Sets the state of the button - true if the button is selected, otherwise sets it to false.

24 What is the use of isSelectedIndex( ) method in JList control?

**Ans:** This method is used to check whether the index specified in the parenthesis has been selected or not.

The syntax of this method is given below:

Syntax:

```
jList.isSelectedIndex(int num);
```

---X---

### OUTPUT FINDING QUESTIONS

1 Write the output following code when executed:

(i) System.out.println("Hello".charAt(3));

(ii) System.out.println("Good morning".substring(4));

**Ans:** (i) l (ii) morning

2. Write the output of the following code :

```
int x , y = 0;
```

```
for(x=1;x<=5;++x) y = x++;
```

```
--y ;
```

**Ans:** 7 4

3. Find the output of the code:

▲ --- 24

**Ans:** 24

```
int f=1,i=2; do
{ f*=i;
}while(++i<5); System.out.println(f);
```

4. What will be the value of j and k after execution of the following code:

```
int j=10,k=12; if(k>=j)
{
k=j; J=k;
}
```

**Ans:** 10 10

5. How many times, the following loop gets executed?

```
i=0;
while (i> 20)
{
//Statements
}
```

**Ans:** 0 times

6. How many times, the following loop gets executed? i=0;

```
do
{
//Statements
}while (i> 20);
```

**Ans:** 1 time

7. What will be the contents of jTextField1 and jTextField2 after executing the following statement:

```
StringBuffer s= new StringBuffer("Common Wealth");
int c= s.capacity();
s.insert(0,'E'); s.reverse(); jTextField1.setText(""+c);
jTextField2.setText(s.toString());
```

**Ans:** 29

htlaeWnommoCE

```
int First = 7;
int Second = 73; First++;
if (First+Second> 90)
jlabel1.setText("value is 90 ");
else
jlabel1.setText("value is not 90 ");
```

**Ans:** value is not 90

9. Find the output

```
int Number1 = 7, Number2=8;
int Second = 73;
if (Number1>0 || Number2>5)
if (Number1>7)
jTextField1.setText("Code Worked");
else
jTextField1.setText("Code MightWork");
else
jTextField1.setText("Code will not Work");
```

**Ans:** Code MightWork

10. How many times will the following loop get executed?

```
x = 5;
y = 36;
while ( x <= y) { x+=6; }
```

**Ans:** 6

11. What will be the content of the JTextArea1 after executing the following code?

```
Int Num = 1;
do{
jTextArea1.setText(Integer.toString(++Num) + "\n");
Num = Num + 1;
}
while(Num<=10)
```

**Ans:** 10

12. What will be the contents of jTextField1 and jTextField2 after executing the following code:

```
String s="KENDRIYA VIDYALAYA GUNA"  
jtextfield1.setText(s.length()+" ");  
jtextfield2.setText(Math.round(2.34)+"");
```

**Ans:** 23 2

13. What will be the value of s after executing the following code? double i,sum=2

```
for(i=3;i<8;++i)  
{  
    if(i%4==0)  
    {  
        break;  
        sum=Math.pow(sum,i);  
    }  
    else  
        sum+=i/2;  
}
```

**Ans:** 150.0625

14. What will be the content of jTextField1 and jTextField2 after executing the following code:

```
String st="New to Information Technology";  
jTextField1.setText(st.replace("Technology","Practices");  
jTextField2.setText(st.substring(7));
```

**Ans:** New to Information Practices

Information Technology

18. Predict the output for tan & tan1 if sac equals 7?

```
int tan = 0, tan1 = 4 ;  
if ( sac == 2 )  
{  
    tan = 4 ; tan1 = 0;  
}  
else
```

```
if (sac == 8)
{ tan = 0 ; tan1 = 4; }
JOptionPane.showMessageDialog( null , “ tan = “ + tan +” , tan1 = “ + tan1 ) ;
```

**Ans:** tan = 0 tan1=4

19. Give the output for the following code fragment:

```
v = 20;
do
{
JOptionPane.showMessageDialog( null , v + “ ” ) ;
}while ( v< 50 ) ; JOptionPane.showMessageDialog( null ,“Bye“ ) ;
```

**Ans:** Infinite loop

20. What will be the output produced by following code fragment?

```
float x=9;
float y=5;
int z=(int)(x/y);
switch(z)
case 1:
x=x+2;
case 2:
x=x+3;
default:
x =x+1;
}
System.out.println(“value of x:”+x);
```

**Ans:** 15

24. What will be the contents of jTextField1 and jTextField2 after executing the following code:

```
String s = “Sun Micro Systems”;
jTextField1.setText(s.length()+””);
jTextField2.setText(s.toLowerCase());
```

**Ans:** jTextField1 : 17 jTextField2 : abc micro systems

25 What values will be assigned to the variable ua ,ub, uc and fail after execution of the following

program segment:

```
int i=0,ua=0,ub=0,uc=0,fail=0;
```

```
while(i<=5)
```

```
{
```

```
switch ( i++ )
```

```
{
```

```
case 1 :
```

```
++ua;
```

```
case 2 :
```

```
++ub;
```

```
uc++;
```

```
break;
```

```
case 3 :
```

```
case 4 :
```

```
++uc;
```

```
ua++;
```

```
ub++;
```

```
break;
```

```
default :
```

```
++fail;
```

```
} }
```

**Ans:** ua=1 ub=1 uc=0

### ERRORS FINDING QUESTIONS

1. Rewrite the following code after removing the errors(if any). Underline the corrections.

```
int i=2,j=5;
```

```
while j>i
```

```
jTextField1.getText(j is greater);
```

```
j--; ++i;
```

```
}JOptionPane.showMessageDialog>Hello);
```

**Ans:**

```
int i=2,j=5;
while( j>i)
{
jTextField1.getText("j is greater");
j--; ++i;
}
JOptionPane.showMessageDialog("Hello");
```

2. Rewrite the code after removing the errors (if any) and underline the corrections.

```
int sum,value,inct;
int i
for(i= =0;i=10;i++)
sum=sum+i;
inct++;
```

**Ans:**

```
int sum,value,inct;
int i;
for(i=0;i<=10;i++)
sum=sum+i;
inct++;
```

3. Rewrite the code after removing the errors (if any) and underline the corrections.

```
int y=3;
switch(y);
{
case 1:
System.out.print("Yes its One");
case2:
System.out.println("Yes its more than Two");
break;
case else:
System.out.print("Invalid Number);
```

**Ans:**

```
int y=3;
switch(y)
{
case 1:
System.out.print("Yes its One");
break;

case 2:
System.out.println("Yes its more than Two");
break;
default: System.out.print("Invalid Number");
}
```

4. Rewrite the code after removing the errors (if any).

```
int x = 0;
int n= Integer.parseInt(Jlabel1.getText());
```

**Ans:**

```
int x = 0;
int n= Integer.parseInt(JLabel1.getText());
```

5. Rewrite the code after removing the errors (if any). M=1, N=0;

```
For(;m+n<19;++n) System.out.println("hello");
M=m+10;
```

**Ans:**

```
m=1; n=0;
for(;m+n<19;++n) System.out.println("hello");
m=m+10;6.
```

Rewrite the code after removing the errors (if any).

```
int y=6,p;
do{
y=3.14*y;
p=y%10;
if p=2 System.out.print("Two");
```

```
while(y>1)
```

**Ans:**

```
int y=6,p;
```

```
do
```

```
{
```

```
y=3.14*y;
```

```
p=y%10;
```

```
if (p==2)
```

```
System.out.print("Two");
```

```
}while(y>1);
```

7. Rewrite the following program code using a for loop:

```
int i,sum=0;
```

```
while(i<10)
```

```
{
```

```
sum +=i; i+=2;
```

```
}
```

**Ans:**

```
int i, sum=0;
```

```
for(i=0;i<10;i+=2)
```

```
{
```

```
sum +=i;
```

```
}
```

8. Rewrite the following code using while loop :

```
int i,j;
```

```
for(i=1;i<=4;i++)
```

```
{
```

```
for(j=1;j<=i;j++)
```

```
{
```

```
System.out.print(j);
```

```
}
```

```
System.out.println( );
```

```
}
```

**Ans:**

```
int i=1,j;
```

```
while(i<=4)
```

```
{
```

```
  j=1;
```

```
  while(j<=i)
```

```
  {
```

```
    System.out.print(j);
```

```
    ++j;
```

```
  }
```

```
  i++;
```

```
  System.out.println();
```

```
}
```

9. Rewrite the following code using

while loop

```
int i,j;
```

```
for(i=1,j=2;i<=6;i++,j+=2)
```

```
System.out.println(i++);
```

```
System.out.println("Finished!!!");
```

**Ans:**

```
int i=1,j=2;
```

```
while(i<=6)
```

```
{
```

```
  System.out.println(i++);
```

```
  i++; j+=2;
```

```
}
```

```
System.out.println("Finished!!!");
```

10. Rewrite the following code using for loop.

```
int i=0; while(++i<20)
```

```
{  
if( i==8)  
  
break;  
System.out.println(i++);
```

**Ans:**

```
}  
int i;  
for(i=1;i<20;++i)  
{  
if( i= =8)  
break;  
System.out.println (i++);  
}
```

11. Write the equivalent switch case for the following code:

```
if (num1 == 1 )  
jTextField1.setText("Number is one");  
else If (num1 == 2 )  
jTextField1.setText("Number is two");  
else If (num1 == 3 )  
jTextField1.setText("Number is three");  
else  
jTextField1.setText("Number is more than three");
```

**Ans:** Switch(num1)

```
{  
case 1 :  
jTextField1.setText("Number is one");  
break;  
case 2 :  
jTextField1.setText("Number is two");  
break;
```

case 3 :

```
textField1.setText("Number is three");
```

```
break;
```

```
default:
```

```
textField1.setText("Number is more than three");
```

```
}
```

12. Rewrite the following code fragment using

switch :

```
if(ch == 'E') east++;
```

```
if(ch == 'W')
```

```
west++; if(ch == 'N')
```

```
north++; if(ch == 'S')
```

```
south++;
```

```
else
```

```
OptionPane.showMessageDialog(null, "unknown");
```

**Ans.** switch(ch)

```
{
```

```
case 'E':
```

```
east++;
```

```
break;
```

```
case 'W':
```

```
west++;
```

```
break;
```

```
case 'N':
```

```
north++;
```

```
break;
```

```
case 'S':
```

```
south++;
```

```
break;
```

```
default :
```

```
jOptionPane.showMessageDialog(null, "unknown");
```

```
}
```

13. Rewrite the following code using for loop:

```
int i = 0;
```

```
while(++i < 20)
```

```
{
```

```
if(i == 8)
```

```
break;
```

```
System.out.println(++i);
```

```
}
```

**Ans:** for(int i = 0; ++i < 20;)

```
{
```

```
if(i == 8)
```

```
break;
```

```
System.out.println(++i);
```

```
}
```

```
private void btnResActionPerformed
```

```
(java.awt.event.ActionEvent evt)
```

```
{
```

```
// TODO add your handling code here:
```

```
int num = Integer.parseInt(tNum.getText());
```

```
if(num%2==0) lRes.setText("Even Number");  
else lRes.setText("Odd Number");  
}
```

2 Greater of Two numbers:

```
private void  
jButton1ActionPerformed(java.awt.event.ActionEvent evt)  
{  
    // TODO add your handling code here:  
    int m = Integer.parseInt(txtM.getText());  
    int n = Integer.parseInt(txtN.getText());  
    if(m>n) lblRes.setText("Greater is:" + m);  
    else if(n>m) lblRes.setText("Greater is:" + n);  
    else lblRes.setText("Both are equal");  
}
```

3 Greatest among three un-equal numbers:

```
private void  
jButton1ActionPerformed(java.awt.event.ActionEvent evt)  
{  
    // we are not taking into account the cases of equality here:  
    int x = Integer.parseInt(txtX.getText());  
    int y = Integer.parseInt(txtY.getText());  
    int z = Integer.parseInt(txtZ.getText());  
    if(x>y && x>z) lblRes.setText("Greatest:" + x);  
    else if(y>z) lblRes.setText("Greatest:" + y);  
    else lblRes.setText("Greatest:" + z);  
}
```

4 Week Day Application that prints the week day depending upon the weekday number

```
private void  
btnResActionPerformed(java.awt.event.ActionEvent evt)  
{  
    int daynum = Integer.parseInt(txtNum.getText());  
    String day;  
    switch(daynum)  
    {  
        case 1:  
            day="Monday";  
            break;  
        case 2:  
            day="Tuesday";  
            break;  
        case 3:  
            day="Wednesday";  
            break;  
        case 4:  
            day="Thursday";  
            break; case 5:  
            day="Friday";  
            break;  
        case 6:  
            day="Saturday";  
            break;  
        case 7:  
            day="Sunday";  
            break;
```

default:

```
day="Invalid day number";
```

```
}
```

```
lblRes.setText(day);
```

5 Grade calculation based on marks scored out of 100

90 to 100 - A 75 to <90 - B 60 to <75 - C 45 to <60 - D 33 to <60 - E 0 to <33 - F other invalid score private void

```
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
```

```
{
```

```
// TODO add your handling code here:
```

```
int score = Integer.parseInt(txtNum.getText());
```

```
String grade;
```

```
if(score>=90 && score<=100) grade="A";
```

```
else if(score>=75 && score<90) grade="B";
```

```
else if(score>=60 && score<75) grade="C";
```

```
else if(score>=45 && score<60) grade="D";
```

```
else if(score>=33 && score<45) grade="E";
```

```
else if(score>=0 && score<33) grade="F";
```

```
else grade="Invalid score";
```

```
lblRes.setText(""+grade);
```

```
}
```

6 Check whether an input character is Alphabet, digit or special character

```
private void  
jButton1ActionPerformed(java.awt.event.ActionEvent evt)  
{  
    char ch = txtChar.getText().charAt(0);  
    String res;  
    if(Character.isLetter(ch))  
        res = "Alphabet"; else if(Character.isDigit(ch))  
        res = "Digit";  
    else  
        res = "Special Character";  
    lblRes.setText(res);  
}
```

7 The following interface is used to calculate the net salary of a person after deducting the tax from the gross based upon the selection from the available radiobutton options (10%, 20% and 30%). The click event of the button will display the tax and the net salary.

8. The following interface is used to calculate the net salary of a person after deducting the tax from the gross based upon the selection from the available radiobutton options (10%, 20% and 30%). The click event of the button will display the tax and the net salary.

```
private void  
btnGenActionPerformed(java.awt.event.ActionEvent evt)  
{  
    // TODO add your handling code here:  
    int gross = Integer.parseInt(txtSal.getText());  
    int tax, net; if(rbTe.isSelected())  
        tax = gross*10/100;  
    else if(rbTw.isSelected())  
        tax = gross*20/100; else tax = gross*30/100;  
    net = gross - tax; lblTax.setText(""+tax);  
    lblNet.setText(""+net);  
}
```

8 The following interface calculate the fees of a student based upon the given criteria. For a boy the fees is 1000. For a girl the fees is 500. For a single girl child as selected from the check box there is no fees. Write code under the Action event of the button to display the name and fees in a JTextArea as shown.

```

private void
btnResActionPerformed(java.awt.event.ActionEvent evt)
{
String name = txtName.getText(); int fee; if(rbtG.isSelected())
{
fee = 500; if(chkSg.isSelected()) fees=0;
}
else fees=1000; txtRes.append("Name:"+name+"\n");
txtRes.append("Fees:"+fee);
}

```

9 The math editor interface is shown below. It performs the Add, Subtract and Multiply operations based on the selections from the JComboBox. Write code under the get Result button to achieve the desired results.

```

private void
btnResActionPerformed(java.awt.event.ActionEvent evt)
{
int choice = cmb1.getSelectedIndex();
//index for the elements from 0 to n-1
int n = Integer.parseInt(t1.getText());
int m = Integer.parseInt(t2.getText());
switch (choice)
{
case 0:
lblRes.setText(""+(m+n));
break;
case 1:
lblRes.setText(""+(m-n));
break;
case 2:

```

```
lblRes.setText(""+(m*n)); break;
}
```

10 The following application is used to add, delete and display selected element from a `JList` box (lName) at run time. Specific codes are written under each button to achieve this.

#### Step 1:

To make it possible to use the dynamic feature of `JList` box we must import the following:

```
import javax.swing.DefaultListModel;
```

#### Step 2:

Declare an object of `DefaultListModel` class as the first statement in

```
JFrame
class: public class
ListApp extends
javax.swing.JFrame
{
DefaultListModel listModel;
```

#### Step 3:

Instantiate the `listModel` object inside the constructor of the

```
JFrameC
class: public
ListApp()
{
//this is the already available inside the JFrame code window
initComponents();
// this is already available
listModel = new DefaultListModel();
lName.setModel(listModel);
}
```

#### Step 4:

Adding elements in the `JList` box:

```
private void
btnAddActionPerformed(java.awt.event.ActionEvent evt)
{
// TODO add your handling code here:
```

```
listModel.addElement(txtName.getText());  
}
```

#### **Step 5:**

Deleting the selected element from the jListBox:

```
private void  
btnDelActionPerformed(java.awt.event.ActionEvent evt)  
{  
    // TODO add your handling code here:  
    listModel.remove(lstName.getSelectedIndex());  
}
```

#### **Step 6:**

Displaying the selected element in a

jLabel:

```
private void  
  
btnShowActionPerformed(java.awt.event.ActionEvent evt)  
{  
    // TODO add your handling code here:  
    lblRes.setText((String) lstName.getSelectedValue());  
}
```

11 The following application retrieves the password from a jPasswordField and displays it in a JLabel

```
private void  
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
```

```

{
// TODO add your handling code here:
String pw = new String(txtPw.getPassword());
lblPw.setText(pw);
}

```

12 The following application demonstrates the use of JOptionPane for displaying message and for accepting input from a user.

```

private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
// TODO add your handling code here: javax.swing.JOptionPane.showMessageDialog(rootPane,"Hello World", "My
Caption", 1);
}

```

```

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt)
{
// TODO add your handling code here:
String str = javax.swing.JOptionPane.showInputDialog("Enter a string");
lblRes.setText(str);
}

```

13 The following application displays the series of even and odd natural numbers depending upon the selected option from the JRadioButton

```

private void
btnShowActionPerformed(java.awt.event.ActionEvent evt)

```

```

{
int limit = Integer.parseInt(txtNum.getText());
if(rbtEven.isSelected())
{
for(int i=2;i<=limit;i+=2)
txtRes.append(""+i+" ");
}
else
{
for(int i=1;i<=limit;i+=2)
txtRes.append(""+i+" ");
} }

```

14 The following interface computes the sum of n natural numbers and factorial of a number as entered by user depending upon the selection of the jRadioButton:

```

private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
int limit = Integer.parseInt(txtNum.getText());
int sum = 0, fact = 1;
if(rbtSum.isSelected())
{
for(int i=1;i<=limit;i++)

```

```
sum=sum+i;
lblRes.setText(""+sum);
}
else
{
for(int i=1;i<=limit;i++)
fact=fact*i;
lblRes.setText(""+fact);

} }
```

15 The following interface checks whether an input number is prime or composite.

```
private void
btnCheckActionPerformed(java.awt.event.ActionEvent evt)
{
int num = Integer.parseInt(txtNum.getText());
boolean prime = true;
for(int i=2;i<=num/2;++i)
if(num%i==0)
{
prime=false;
break;
}
if(prime==true)
lblRes.setText("Prime Number");
else
lblRes.setText("Composite Number");
```