UNIT II(PROGRAMMING IN JAVA AND WEB APPLICATION DEVELOPMENT)

Answers To One Mark Questions

1. <BODY BACKGROUND= "CLOUDS.JPG">

2. int x=0;

   int n= Integer parseInt(Jlabel1.getText());

3. Ms kiran should set **visible** property of jTextField as true.
   jTextField.setVisible(true);

4. **break** statement is used to terminate the loop whereas continue statement is used to skip the current iteration and take the iteration value.
5. XML is extensible markup language.

HTML	**XML**
Both container as well tags are available	Only container tags are supported in tags
It is not case sensitive	It is case sensitive
All tags are predefined	No predefined tag is available in XML. Only user defined tags are supported.

6. dblMarks=Double.parseDouble(strMarks);

7. How
   r U?
8. New to Information Practices
   Information Practices

9. Ranjeeta should use **model** property to add items in list box.

10. When none of the case in switch statement will match, control will transfer to **default** clause.

11. Properties of jTextField are : **text, editable, enabled**

12. 7

13. Army
    Public School Asr

14. Rapid Application Development (RAD), is a software programming technique that allows quick development of software applications. It was developed by James

Martin in the 1980s and introduced in the 1990s.

15. int, float

16. IDE – Integrated Development Environment
    JVM – Java Virtual Machine

17. Variable is user defined whereas; reserve word is predefined. Variable can be modified but reserve word cannot be modified.

18. Variables in java are declared by using the following syntax:
    datatype var;
    Example is:
    int rate;

19. jLabel is uneditable at runtime whereas; jTextField is by default editable at runtime.

20. 15

21. When a derived class has a function with same name and signatures as in the base class, it is known as method overriding.

22. It will display: **nfor**

23. longPhone= Long.parseLong(strphone);

24. **<TABLE>** for making table and **<TR>** to add rows in the table.

25. <OL> is numbered (ordered) list. <UL> is bulleted (unordered) list.

26. **null**

27. Ms khorans should use **if** statement. **if** statement can manage equality as well as non equality based conditions wheras; switch statement can manage only equality based conditions.

28. jTextField1.setEditable(false);

29. 14

30. 212

31. 162

32. Aa

33. Keyword **extends** is used to call Parent Class property in a Child Class.

34. <HR>

35. An **abstract** class is that class which cannot be instantiated and works as foundation class.

36. An **abstract** method is that method which has only declaration and no definition.

37. **model property.**

38. **private** members are never inherited by derived class whereas; **protected** members are inherited by derived class.

39. dblMarks = Double.parseDouble(strMarks);

40. Keyword **super** is used to access the base class constructor and overridden function.

41. Unicode is 16 bits character set used by java.

42. Escape sequence "\n" is for new line character and "\t" is used for tab space.

43. The keyword "**this"** refers to currently calling object. It is automatically created and initialized by java. So, you can refer to current object by using keyword **this.**

44. **nfor**

45. JPanel, JToolbar, JFrame are some examples of swing ontainer.

46. Value of **k** will be 12 and **j** will be 12

47. Elements that control the execution of loop are :
initial value , condition (test expression) and updation.

48. Polymorphism also known as function overloading. Writing more than one function with same name but with
(i) different number of parameters.
Or (ii) different type of parameters.

49. setText(); method is used to set the value in jLabel.
    Example is: jLabel1.setText("INDIA");

50. By setting the **editable** property of jTextField as flase (unchecked)
    jTextField.setEditable(false);

51. Method **getSelectedValue();** is used.

52. An "**if statement** "can manage equality based as well as non-equality based test expressions whereas; switch can manage only equality based test expressions. All programs written using **switch** statement can be converted into **if** statement and not vice versa. So "if statement" is more general than "switch statement".

53. Keyword **void** is used with function declaration/definition denotes that function will not return any value to the calling function.

54. int big;
    int A=Integer.parseInt(jTextField1.getText());
    int B=Integer.parseInt(jTextField2.getText());
    big= (A>B)? A:B;
    jTextField3.setText(""+big);

55. Return type of **toUppercase()** is **String** and return type of **sqrt()** is **double**

56. An **interface** is a collection of constants (final variables) and abstract methods. Keyword **interface** is used to declare an interface.

57. Infinite times.

58. jList1.getSelectedIndex();

59. **model** property.

60. Conditional operator (**?**) also known as Ternary operator (**?**) works on three operands. It can be used in place **if else** statement. This operator is used to check a condition and produce logical result; either True or False. The syntax is:
    Var= condition ? expression1 : expression2;

61. Return type of method pow() is **double** and return type of method toLowerCase() is String.

62. Container control can contain other child controls in it. e.g. **jPanel**.

63. Constructor function
    (i) has same name as that of class
    (ii) has no return type not even void

64. Keyword **void** in a method definition denotes that function will not return any value.

65. 5 times.

66. **append();** method will be used.

67. **TextField.setText("break");** will never execute.

68. Inheritance is the act of deriving a new class from an existing one. Types of Inheritance are: Single level, Multilevel, Multiple, Hierarchical and Hybrid.

69. smarks + " "; or Double.toString(smarks);

70. **Multiple inheritance** is not supported by java.

71. Source code is stored in **.java** file. When **.java** file is compiled, bytecode is generated and stored in **.class** file.

72. **float, double**

73. Property window is used to set the properties(features such as name, appearance etc.) of various swing controls.

74. Integer.parseInt("345");

75. Operator "=**"** is an assignment operator used to assign values to variable. == is a relational operator used to compare the variables/values.

76. 2746.55

77. APS
Half Yearly Exam

78. The '**default**' section of switch statement is used to execute statement(s), when none of the specified cases mentioned match.

79. Open the HTML file using any web browser / in preview mode of web design tool.

80. x= 500;
x+= 50;

y=x;
OR
x=500;
x=x+50;
y=x;

81. jComboBox.

82. "if" statement is used to select among two alternatives whereas; switch statement is used to select among multiple alternatives. "if" statement can handle equality based as well as non-equality based comparisons whereas; switch statement can handle on equality based conditions.

83. **ButtonGroup** control should be used so that only one of the radio buttons (Male and Female) can be selected at a time.

84. **parseInt()** method converts a string to an integer.

85. <H1> tag is used to display text with largest font size so as to act as a heading.

86. In **FocusLost()** event handler of AmountTF text field.

87. System.exit(0);

88. XML is case sensitive so <EMAIL> and <email> are not same. Thus the ending tag should be <EMAIL>

89. doClick() method .

90. 14

91. recSet.absolute(8);

92. Both **Line Wrap** and **Word Wrap** properties set to true will help him achieve this.

93. This mechanism is known as **Fall-through**.

94. (i) " "+ K; (ii) String.valueOf(K);

95. rs.first();

96. Sequence is : Initialization , testing , execution of loop body, updation

97. <P> tag

98. <UL Type=Square>

99. echochar property of jPasswordField will be set as “$”.

100. An empty or null statement is semicolon (;) only. It is used when syntax demands a statement but the logic does not.

101. **null** is a literal, in the same sense that false, 10, and '\n' are literals. It's not a "**keyword**", technically, but it is a character string that is treated specially by the compiler if the compiler encounters it in a **java** source file. So, you cannot name a variable "**null**".

Answers To Two/Three Marks Questions

1. <P> tag is paragraph tag.

<P> tag	** tag**
It is container tag	It is empty tag
It can take attributes	No attribute is given
It leaves one blank line before start of new paragraph	It leaves no blank line

2.

HTML	**XML**
Both container as well tags are available	Only container tags are supported in tags
It is not case sensitive	It is case sensitive
All tags are predefined	No predefined tag is available in XML. Only user defined tags are supported.
It Focuses on presentation	It Focuses on data

3. **Similarity**: Both are entry control (pre-tested) iteration statements.

Difference: In case of **while** statement **initialization**, **condition** and **updation** steps are separately defined. Whereas, in case of **for** statement **initialization**, **condition** and **updation** steps are defined in a single row separated by ; (semicolon). Coding size in case of **for** loop is compact.

4. Value of **x** will be 11 and loop will execute 3 times.

5. switch(num1)
```
{
    case 1:  jTextField1.setText("Number is one");
    break;
   case 2:  jTextField1.setText("Number is two");
    break;
    case 3:  jTextField1.setText("Number is three");
    break;
   default: jTextField1.setText("Number is more than three");
 }
```

6. Value of **i will be 4** and value of **sum will be 3.0**

7. <UL> Unordered bulleted list
   <TITLE> To assign the title to web page
   <SUP> For superscript
   <P> To start a new paragraph

8. Inheritance means defining a new class (derived class) in terms of an existing class (base class).
   Derived class will inherit the properties of base class. **Muliple inheritance** is not supported by java.

9. 1 3 5 7 9 11 13 15 17 19

10. int y=3;
```
switch(y)
 {  case 1: System.out.print("Yes its One");
    case 2: System.out.println("Yes its more than Two");
         break;
default: System.out.print("Invalid Number):
 }
```

11.

Ans. Value of **x will be 7** and value of **y will be 4**

12.
```
public void fun(float P, float R, float T)
  {
  float si= P*R*T/100;
  JOptionPane.showMessageDialog(null, si);
  }
```

13.
```
switch(k)
        {
```

```
case 1: Day=" Monday";  break;
case 2:  Day=" Tuesday";  break;
case 3:  Day=" Wednesday";  break;
 default: Day=" -";  break;
 }
```

14.
```
int i,sum=0;
for(i=1=1; i<10; i+=2)
 {
  sum+=i;
  }
```

15.
```
int i , j=5;
i=j+5;
if(i==j)
{
jtextfield1.setText("I and j are unequal");
_____
}
else
jtextfield1.setText("I and j are equal");
```

16.
```
25
 2
```

17. In jTextField whatever text we type is visible as it is but in case of jPasswordField whatever we type is visible in encrypted form set by using the **echochar** property . jTextField is used to enter the data to perform some calculations etc. whereas; jPasswordField is used to enter the password to check the authenticity of user. In case of jTextField, **getText()** method is used to read the data and in case of jPasswordField, **getPassword()** method is used.

18. Method **getText()** is used to read the data entered in the jTextField, jLabel etc. whereas; **setText()** method is used to display the in the jTextField, jLabel etc.

19.
```
I
Like

Java   Programming
```

20. Java uses **Unicode** character set. Unicode is a two byte (16 bits) character code set that has characters representing almost all characters in almost all languages and writing systems around the world including English, Arabic, Chinese and many more.

21.
```
if(code== 'A')
    System.out.println("Addition");
 else if(code== 'S')
    {System.out.println("Subtraction");
    System.out.println("Multiplication");}
  else if (code=='M')
     System.out.println("Multiplication");
 else if(code=='D')
      System.out.println("Division");
  else
          System.out.println("Exponentation");
```

22.

jListBox	jComboBox
It displays the items along with scroll bar.	Combo of jTextField and drop down list with no scroll bar.
getSelectedValue() method is used to read the selected item from jListBox.	getSelectedItem() method is used to read the selected item from jComboBox.

Common property of both the controls i:s **model.**

23.
```
i=1;
while(i<=10)
{
    jTextArea1.append(i+" \n");
    i++;
}
```

24. P= 8 Q= 10500

25. Loop will execute **5 times.** It is **Entry Control (Pre Tested Loop).**

26.
```
public void fun(int Num)
{
int p=1,d;
 while(Num>0)
  {
   d= Num % 10;
   p = p * d;
   Num = Num / 10;
  }
System.out.println(p);
}
```

27. An "**import" statement** is used to inherit extra features, which are not available in java.lang (the default header file) to be used in the program.
import javax.swing.JOptionPane;
To use JOptionPane dialog box in the program.

28. "**if"** statement is a selection statement also called decision making used to execute instruction(s) depending upon test expression. i**f , if else, if else if** (ladder if), **netsted if (**if within if**)** are different forms of if statement.

29. **Scope** means area of the code where the variable is accessible or visible. Variables opened in particular block can be accessed only that block, are called local variables. Global variables are accessible to all blocks.
Life time of a variable means for how much time variable remains open in the memory. Lifetime of local variable is short (limited to a block) whereas; lifetime of a global variable is more.

30. This action involves two steps:
    (i) Loading the JDBC driver by using Class.forName()
    (ii) Making a connection

31. JDBC performs following tasks:
    (i) establishing a connection with a database
    (ii) sending SQL statements to database server
    (iii) processing the result obtained

32. A **result set** refers to a logical set of records that are fetched from the database by executing a query.

33.
10 19
19 12
12 17
17 14

34.
```
int x=0,i, j;                    // variables i and j were not declared
 for(i=1; i<10; i++)
        {
                for( j=5; j>i ; j--)        // variable j was declared as local variable
                {
```

```
                System.put.print("The j is " + j)
                System.out.printLn("The i is " + i);
                }
        x=i+j;  // variable j was declared as local variable (in inner loop) so was not
                //accessible here
         System.out.print("The x is " + x)
          }
```

35. <CENTER> tag to horizontally center aligns the text in a web page.
    <HR> tag Horizontal rule to draw a Horizontal line in a web page.

36. Ans. float cost=Float.parseFloat(jTextField1.getText());
    int nop= Integer.parseInt(jTextField2.getText());
    float tamt= cost * nop;
    jLabel1.setText(" " +tamt);
    float tax = 0.15 * tamt;
    jLabel2.setText(" " +tax);

37. (i) int principal = Integer.parseInt(prinTf.getText());
    (ii) int time = Integer.parseInt (timerTF.getText());
    (iii) float interest = principal * 0.08 * time; //int is reserve word
    (iv) JOptionpane.showMessageDialog(null,"Interest is"+ interest);
    //int is reserve word

38. int i,sum=0;
    for(i=1; i<10; i+=2)
    sum = sum + i;

39. Value of k = 10 and j = 10

40. F**unction overloading** means writing many functions in a single program with same name but with
    (i) different number of parameters
    or (ii) different type of parameters.

    Function overriding means writing function in derived class with same name and signatures as in base class. That is derived class has overridden the functionality of base class function.

41. int R, w=90;
    while (w>= 60)
    {

```
R-=50;
 w=R;
}
```

42.
```
int i;
for( i=0; ++i<20 ; i++)
{
 if( i==8)
 break;
 System.out.println(i);
 }
```

43.
```
int y=3;
switch(y)
{  case 1: System.out.print("Yes it is One");
           break;
  case 2: System.out.println("Yes it is more than Two");
           break;
  default : System.out.print("Invalid Number):
  }
```

44. java bytecode is a low level representation of a java source code program. The java compiler translated the source code into bytecode, which can then be executed using java interpreter. Java Virtual Machine (JVM) is the interpreter to execute the bytecode.

45. Characteristics of java
    - The programs written in java are portable in a network.
    - The code is robust
    - Java is object oriented
    - .java, .class and .jar files created in java are virus free.

46. Identifiers are fundamental building blocks of a program and are used as the general terminology for the names given to different parts of the program viz. variables, classes, functions etc.
    Valid identifiers: My_Pwd_1997 , a20; principal etc.

47. A String is one or more characters enclosed in double quotes considered as a single value.
```
String name = "Ravi Sharma";
String DOB = "1997-12-15";
```

48. **import java.awt.Font;** statement is used to use various font features such as **bold**,**italic ,color** etc. along with **size** and manage the display of output in more attractive manner.
import javax.swing.ImageIcon; statement is used to display desired image in jLabel by writing the code.

49. Ternary operator (?) also known as conditional operator (?) works on three operands. It can be used in place **if else** statement. This operator is used to check a condition and produce to logical result, either True or False. The syntax is:
var=condition ? expression1 : expression2;
e.g. big=(A>B)? A : B;

50. Operator precedence determines how an expression gets evaluated. It is a set of rules that establishes which operators get evaluated first and how operators within the same precedence level associate.

51.
```
int i=0,x=0;
while(i<10)
  {
   if(i%2==0)
   x= x+i;
  System.out.println(x+1);
  }
  I+=1;
}
```

52. Loop will execute **2 times.** It is entry control loop.

53.
```
int I,j;
for(i=1;i<=5;i++)
{
   for(j=1;j<=i;j++)
   jTextArea1.append("*");
    jTextArea1.append("\n");
}
```

54.
```
int rev=0, num,d;
num= Integer.parseInt(t1.getText());          // jTextField named as t1
while(num>0)
```

```
 {
d= num % 10;
rev= rev*10 + d;
 num = num / 10;
}
t2.setText(Integer.toString(rev));                    // jTextField named as t2
```

55.
```
int i=1;
while(i<10)
{
 System.out.println(i);
  i++;
  }
```

56.
```
int num=Integer.parseInt(jTextField1.getText());
 if(num==10)
 jLabel1.setText("Ten");
 else if(num==20)
 jLabel1.setText("Twenty");
 else if(num==30)
 jLabel1.setText("Thirty");
 else
 jLabel1.setText("Others");
```

57. Loop will execute **11 times.**
Output is: **0 1 2 3 4 5 6 7 8 9 10**

58. X ² + Y ² + 2XY

59. Value of **i will be 8** and **j will be 5.**

60.
```
for(i=2; i<51; i+= 2)
{
]OptionPane.ShowMessageDialog (null,i+ "");
}
```

61.
```
i=2; j=5;
 while( j>i) {
  jTextField1.setText("J is greater");
  j--;
  i++;
   }
```

JOptionPane.showMessageDialog(null,“Hello”);

62. (i) Mr Smith
Mr King
Mr Fisher

(ii) Mr King
Mr Fisher

63. (i) Loop will execute infinite times. Value 11 will display infinite times.

64.
```
String pass= new String (jPasswordField1.getPassword());
if(pass.equlas(“hello”))
 new jFrame1().setVisible(true);
```

65.
```
switch(code)
{
case ‘A’:   System.out.println(“Accountant”);
case ‘G’:    System.out.println(“Grade IV”);
            break;
case ‘F’:    System.out.println(“Financial Advisor”);
}
```

66.
```
<HEAD>  <TITLE> Hello World  </TITLE>  </HEAD>
<OL>
    <LI> Short Term Course </LI>
<UL>
    <LI>  <A HREF>  Java with NetBeans </A> </LI>
    <LI>  <A HREF>  MySQL </A> </LI>
 </UL>
```

67. **MAIN()** is the constructor function as it has same name as that of class.
Output is : I am displaying result 26.

68. An **abstract** class is that class which cannot be instantiated, it works as base class. Keyword **abstract** is used **to declare a class as abstract. Concrete** class is derived class of which instance is created.

69. **(i)**Keyword **final** when used with a variable declaration denotes that variable’s values cannot be modified.
Keyword **final** when used with a class declaration denotes that class cannot be extended.
Keyword **final** when used with a method (function) declaration denotes that function cannot be overridden.

(ii) Keyword **implements** is used while implementing (using) an **interface** by a class.
e.g. **class X implements interface1**

70. **JDBC (Java Database Connectivity)** is a framework developed by Sun Java to help java connect to different databases.
ODBC (Open Database Connectivity) is a framework developed by Microsoft to connect to various types of databases.
JDBC provides database connectivity from within Java applications to database whereas; ODBC provides database connectivity to non-java front end applications. ODBC is language independent as it can work with any language but JDBC is language dependent as it works only for java.

71. Elements that control the execution of a loop are:
Initial value, test expression and updation
Syntax: for(initial value; test expression; updation)
{ }
Example:

```
for(n=1; n<=5; n++)
  System.out.println(n);
```

72. 5 times. It is post tested loop statement.

73. (i) <BODY BGCOLOR= "Purple"> (ii) <IMG SRC= "LOGO.GIF">
(iii) <A HREF="HOBBY.HTML"> <IMG SRC= "LOGO.GIF"> </A>
(iv) <OL> <LI> DESKTOPS <LI> LAPTOPS </OL>

74. (i) Keyword **super** is used to access the base class constructor and overridden function.
(ii) Keyword **extends** is used to inherit the properties of base class by derived class.

75.
```
String sname;
sname= JOptionPane.showInputDialog("Enter Name");
JOptionPane.showMessageDialog(null, "Name is"+ sname);
```

76.
```
int firstn,  secondn,N;
firstn  = Integer.parseInt(jTextField1.getText());
secondn = Integer.parseInt(jTextField2.getText());
for(N=firstn; N<= secondn ; N++)
{
    if(N%2==0)
     jTextArea1.append(N + "\n");
 }
```

77. Value of **a will be 5** and Value of **b will be 5**

78. Value of **j will be 11** and Value of k **will be 11**

79. switch(ch)

```
{
case 'E':  east ++ ; break;
case 'W':   west ++ ;break;
case 'N':    north ++ ;break;
case 'S':  south ++ ; break;
default:   west ++ ;
}
JOptionPane.showMessageDialog (null, "unknown");
```

80. Iteration statements also called looping statements are used to execute a set of instructions repetitively. Iteration statements are **while, do while and for.**

81. The variables, which are associated with function name during function call, are called actual parameters.
    The variables (or parameters), which accept the values of actual variables (parameters) in the header line of function definition, are called actual parameters.

82. Scope means area of the code where the variable is accessible or visible. Variables opened in particular block can be accessed only that block, are called local variables. Global variables are accessible to all blocks.
    Life time of a variable means for how much time variable remains open in the memory. Lifetime of local variable is short (limited to control in a block) whereas; lifetime of a global variable is more.

83.

```
int j, s=0;
    for(j=1;(j<10;j+=3)
    {
    System.out.println(x+1)
    s=s+j ;
    }
```

84. Type casting is also known as explicit conversion. It is user-defined that forces an expression to be of specific type . For example, to make sure that the expression t (x+y)/2 evaluates to type float, write it as:

```
 int x,y;
(float) (x+y)/2;
```

85. Output 1 =7.54.5
    Output 2=14.0
    Output 3= 3

Note: System.out.println("Output 4 =" + **7-2**); will generate error.

86.
```
int i,sum=0;
for(i=1;i<15; i=i+3)
{
 sum+=i;
}
```

87. Loop 1 will execute **2 times.** Loop 2 will execute **2 times.**

88. 5
    3

89. i) Mr. Smith
    Mr. King
    Mr. Fisher

    (ii) Mr. King
    Mr. Fisher

90. P=8 Q=10500

91. JVM stands for Java Virtual Machine. JVM is used to interpret the bytecode and execute.

92. jTextField can hold single line of text whereas; jTextArea can hold l multiple lines of text.
    jTextField uses setText() method whereas; jTextArea uses setText() and append() methods.

93. Constant is that quantity of which value does not change during execution of program. A Variable is one which holds certain value, it is a sequence of alphabets and digits but the first character must be alphabet or underscore.

94. An escape sequence is represented by a backslash(\) followed by a character code. It is used to control/ format the display of output.Escape character "\n" is for New Line and "\t" for Horizontal Tab

95. Reserve word also known as keyword are predefined. These words have some special meaning for the language. Example: int, float, while, for, if, else etc.

96. Expression is a combination of operators and operands. Example: X + Y * 7 is an arithmetic expression.

97. Operator "**=**" is an assignment operator used to assign values to variable. == is a relational operator used to compare the variables/values.

98. (i) X=2, Y=0 (ii) X=1,Y=1

99. (i) wrong input (ii) Platform independent
Object oriented
wrong input

100.

(i)
```
jTextField1.setEditable(false);
jTextField2.setEditable(false);
```

(ii)
```
int fee;
jTextField1.setEditable(true);
jTextField2.setEditable(true);
if(joptCom.isSelected())
  fee = 2500;
if(joptArts.isSelected())
fee = 2000;
if(joptMed.isSelected())
fee = 2900;
if(joptNonMed.isSelected())
fee = 2800;
jTextField2.setText(" "+ fee);
```

(iii)
```
int Total_fee;
int fee= Integer.parseInt(jTextField2.getText());
if(jChkBox.isSelected())
      Total_fee = fee+ 200;
else
```

```
        Total_fee = fee;
    jTextField3.setText( " " + Total_fee);
```

(iv) System.exit(0);

101.

(a)
```
txtDiscount.setEditable(false);
txtNetPayable. setEditable(false);
```

(b)

(i)
```
double disc=0, net_amt=0;
int qty= Integre.parseInt(jTextField1.getText());
float price= Float.parseFloat(jTextField2.getText());
float amt= qty*price;
if(jRadioButton1.isSelected())
  disc= 0.12*amt;
  else if((jRadioButton2.isSelected())
     disc=0;
  else if((jRadioButton3.isSelected())
    disc=0.08*amt;
    net_amt= amt-disc;
 if(jCheckBox1.isSelected())
      net_amt= net_amt- 0.05* net_amt;
 jTextField3.setText(""+disc);
 jTextField4.setText(""+net_amt);
```

(ii)
```
txtQty.setText(" ");   txtPrice. .setText(" ");
txtDiscount.setText(" ");          txtNetPayable. setText(" ");
```

(c)
```
System.exit(0);
```

102. Inheritance means defining a new class (derived class) in terms of an existing class (base /super class). Derived class inherit the properties of base (parent) class. Advantage is that, as derived class can use (access) the methods (functions) of base class, so; a large application can be developed in a short period by using already existing code of base class.

103.

(a)
```
rbtnCar.setSelected(true);
```
jtxtRate.setEnabled(false);

(b)
```
public sub jtxtLoanAmtFocusLost(java.awt.event.FocusEvent evt)
{
int Lamt= Integer.parseInt(jtxtLoanAmt.getText());
```

```
            if(Lamt<=0)
              {
                 JOptionPane.showMessageDialog(null,"Invalid Amount");
                jtxtLoanAmt.requestFocus();
          }  }
```

(c)
```
   double p= Double.parseDouble(jtxtLoanAmt.getText());
      double prchgs , intrate ;
       int t=  Integer.parseInt(jcmbYears.getSelectedItem().toString());
       if(rbtCar.isSelected())
        intrate= 14.5;
       else if(rbtnPersonal.isSelected())
        intrate= 13.25;
       else if(rbtnEducation.isSelected())
     intrate=11.75;
prchgs=500;
double CI= p* Math.pow((1+intrate/100) , t) ;
double EMI= (CI+p) /(t*12);
jtxtRate.setText(""+ intrate);
jtxtProcCharge.setText(""+prchgs);
jtxtEMI.setText(""+ EMI);
```

d) System.exit(0);

(e)
```
   jtxtRate.setText("");
       jtxtProcCharge.setText("");
       jtxtEMI.setText("");
      jtxtLoanAmt.setText("");
      rbtCar.setSelected(false);
      rbtnPersonal.setSelected(false);
     rbtnEducation.setSelected(false);
```

104.

(a) readape(), showape(), read(), show()

(b) No member can be accessed.

(c) readhuman(), showhuman(), readape(), showape(), read(),show()

105.

(a)
```
txtDiscount.setEditable(false);
txtNe.setEditable(false);
btnCalNet.setEnabled(true);
```

(b)
```
String model= (String) cmbModel.getSelectedItem();
double price=0.0,discount=0.0;
if(model.equlas("EW SX")
   { price= 16000; discount=  0.10* price;}
if(model.equlas("EW DX")
{  price= 12000; discount=  0.12* price; }
if(model.equlas("EW MX")
{  price= 22000; discount=  0.08* price; }
 If(chkAuto.isSelected())
    price=price + 4000;
   txtTotal.setText(" "+ price);
   txtDisccount.setText(" "+ discount);
  btnCalNet.setEnabled(true);
```

(c)
```
 double price= Double.parseDouble( txtTotal.getText());
 double discount= Double.parseDouble(txtDisccount.setText());
  double net= price – discount;
   txtNet.setText(" "+ net);
```

106.

(a) ButtonGroup Property of radiobuttons should be set.

(b)
```
 FemaleRB.setSelected(true);
  IntermediateCB.setSelected(false);
 GraduateCB. setSelected(false);
PostGraduateCB. setSelected(false);
```

(c)
```
if(PostGraduateCB.isSelected())
{
  GraduateCB. setSelected(true);
 IntermediateCB. setSelected(true);
}
  If(GraduateCB. isSelected())
 IntermediateCB. setSelected(true);
```

107.

(a)
```
txtItem.setText(" ");
```

```
txtPrice. setText(“ ”);
txtDiscount. setText(“ ”);
 txtFA. setText(“ ”);
```

(b)
```
int qty =Integer.parseInt(txtItem.getText());
if(qty<=0)
{
  txtItem.setText(“”);
   JOptionPane.showMessageDialog(nill,”Invalid Data”);
}
```

(c)
```
float price=Float.parseFloat(txtPrice.getText());
 int qty =Integer.parseInt(txtItem.getText());
  float amt= qty*price;
 double Disc=0;
      if(MenRb.isSelected())
      {
      if(amt<10000)
      Disc= 0.30*amt;
      else
         Disc= 0.50*amt;
      }
   else if(WomenRb.isSelected())
      {
      if(amt<8000)
      Disc= 0.40*amt;
      else
         Disc= 0.50*amt;
      }
   else if(KidRb.isSelected())
      {
      if(amt<5000)
      Disc= 0.20*amt;
      else
         Disc= 0.30*amt;
      }
   if(CHCheck.isSelected())
     Disc=Disc+ 0.05*amt;
    double final_amt= amt-Disc;
   txtDiscount.setText(“”+ Disc);
   txtFA.setText(“”+final_amt);
```

108.

(a)
```
TxtAmt.setText("");
TxtDisc.setText("");
TxtNet.setText("");
```

(b) (i)
```
TxtDisc.setEnabled(false);
TxtNet.setEnabled(false);
```

(ii)
```
OptCash.setSelected(true);
```

(c)
```
double amt=Double.parseDouble(TxtAmt.getText());
double disc;
    if(OptCash.isSelected())
     {
     if(amt<10000)
        disc=0.20*amt;
      else
         disc=0.25*amt;
     }
     else if(OptCheque.isSelected())
        {
     if(amt<15000)
        disc=0.10*amt;
      else
         disc=0.15*amt;
     }
     else if(OptCredit.isSelected())
        {
     if(amt<10000)
        disc=0.10*amt;
      else
         disc=0.12*amt;
     }
    double net=amt-disc;
    TxtDisc.setText(""+disc);
    TxtNet.setText(""+net);
```

(d)
```
JOptionPane.showMessageDialog(null,"Thank You");
System.exit(0);
```

109. 1

110. jTextField displays input / output characters as they are. jPasswordField does masking

of keyboard input from user, using an echo character '*' by default.

111. 56
72

112. XML tags are created by the user as there are no standard tags.
For example:
To store name, the tag <name> may be used as:
<name> Sumedha </name>

113.
```
int marks, temperature;
marks= Integer. parseInt(JTextField1.getText());
temperature= Integer. parseInt(JTextField2.getText());
if ( (marks < 80)  &&  (temperature > = 40 ) )
{
    System.out.println9"not Good");
}
else
{
    System.out.println("OK");
}
```

114. 5 times.

115.
```
String tour= "";
int c1 = Integer. parseInt(JTextField1.getText());
if( c1==8)
     tour = " \n You are going to Camp Ramgarh";
else if (c1==9)
    tour = " \n You are going to Manali, Rohtang Pass" ;
else if (c1==10)
     tour = " \n You are going to Chail";
else
    tour= " No School tour for you  this time";
```

116. sum = 12
x=6

117. 4
BEST

118. (i) int Total=0;

```
If ( jCheckBox1. isSelected())
{
JTextField3.setText ( " " +10);
Total= Total + 10;
}
if ( jCheckBox2. isSelected())
{
JTextField4.setText (" "  +10);
Total= Total + 10;
}
if ( jCheckBox3. isSelected())
{
JTextField5.setText (" "  +10);
Total= Total + 10;
}
jTextField6.setText (" " + Total);
```

(ii) jTextField1.setText(" ") ;
jTextField2.setText(" ") ;
jTextField3.setText(" ") ;
jTextField4.setText(" ") ;
jTextField5.setText(" ") ;
jCheckBox1.setSelected(false) ;
jCheckBox2.setSelected (false) ;
jCheckBox3.setSelected (false) ;
jTextField6.setText(" ") ;
Note: **NULL** in place of " " can also be used.

(iii) System.exit(0) ;

119. (i) jTextField4.setText("0");
jTextField5.setText("0");
jTextField4.setEditable(false);
jTextField5.setEditable(false);

(ii) double disc=0.0,addl_disc=0.0;
double bill_amt= Double.parseDouble(jTeaxtField2.getText());
if(jRadioButton1.isSelected())
disc= 0.20* bill_amt;
else if(jRadioButton2.isSelected())
disc= 0.15* bill_amt;
else if(jRadioButton3.isSelected())

```
        disc= 0.10* bill_amt;
      jTextField3.setText(“”+disc);
       if(bill_amt>25000)
      addl_disc =0.05* bill_amt;
     jTextField4. setText(“”+addl_disc);
     jButton2.setEnabled(true);
```

(iii)
```
double bill_amt= Double.parseDouble(jTeaxtField2.getText());
double disc= Double.parseDouble(jTextField3.getText());
double addl_disc= Double.parseDouble(jTextField4.getText());
 double net= bill_amt-disc-addl_disc;
  jTextField5.setText(“”+net);
```

120.

(i)
```
int startNum= Integer.parseInt(jTextField1.getText());
int lastNum= Integer.parseInt(jTextField1.getText());
int i;
for(i=startNum;i<=lastNum;i=i+2)
jTextArea1.append( i +”\t”);
```

(ii)
```
jTextField1.setText(“”);
jTextField2.setText(“”);
jTextArea1.setText(“”);
```

(iii)
```
System.exit(0);
```

121. In switch case statement break statement is used to avoid **fallthrough.** **Fallthrogh** means after executing a particular case, control should transfer out of switch statement; that is only required case statement should execute.

Example:

```
If (i) ch= ‘a’ (ii) ch= ‘x’
      switch(ch)
       {
            case  ‘a’:   System.out.println(“Mr Smith”);

            default:    System.out.println(“Mr King”);
                          break;
       }
       System.out.println(“Mr Fisher”);
```

<u>Output will be:</u>

(i) Mr Smith
Mr King
Mr Fisher

(ii) Mr King
Mr Fisher

122. <OL> tag. Attributes are Type, Start.
Example is <OL Type = "A" Start=2>

123. <A Href= "http://www.cbse.nic.in"> Click here to Connect to CBSE </A>

124. 1 4 9 16

125. int N= Integer.parseInt(jTextField1.gettext());
jTextField2.setText(" "+ (N*N*N));

126. Eligible

127. int C= jComboBox.getSelectedIndex();
if(C==0)
Amount = Bill ; break;
else if(C== 1)
Amount = 0.9*Bill ; break;
else if(C==2)
Amount = 0.8*Bill ; break;
else Amount = Bill;

128. Six times.

129. P will be 5 and Q will be 6.

130. Great
Country
India

131. **getSelectedIndex()** method will be used to extract value of Index of selected item in jComboBox and jListBox.

132. isSelected()method will return true if the current checkbox is selected/checked.
setSelected() method is use to check/uncheck a particular checkbox.

133. getSelectedItem() and getSelectedIndex()

134. getSelectedValue() and getSelectedIndex()

135. <img src=”abc.jpg” **Height=100 Width=100 Border=4**>

136. Loop1 will execute 4 times. Loop2 will also execute 4 times. Loop1 is entry control loop. Loop2 is exit control.

137. 5 in jTextField1 and 11 in jTextField2 will display.

138. (i) Go Green
India
(ii) 6

139.
```
int Last=Integer.parseInt(jTextField1.getText());
int C=1;
while(C<=Last)
{
jTextArea1.setText(Integer.toString(C));
C++;
}
```

140.
```
jButton1.setEnabled(true);
jButton1.setText(“Click Here”);
```

141. In jTextField1 8 and In jTextField2 6 will display.

142. Value of Text1 will be IndiaGood Morning and Text2 will be India

143.

(i)
```
double Basic= Double.parseDouble(jTextField1.getText());
double tax =0;
if(Basic<50000)
tax=0.2*Basic;
else if(Basic>=50000)
tax= 0.3* Basic;
jTextField4.setText(“”+tax);
```

(ii)
```
double Basic= Double.parseDouble(jTextField1.getText());
double HRA= Double.parseDouble(jTextField2.getText());
double DA= Double.parseDouble(jTextField3.getText());
double tax= Double.parseDouble(jTextField4.getText());
double salary=(Basic+ DA + HRA) – tax;
jTextField5.setText(“”+salary);
```

(iii) jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField5.setText("");

144. By setting the **selectionMode** property

145. True

146. jPasswordField does not directly display its contents; instead it uses a single character (usually an asterisk) to represent each character that it contains, so that it is possible to see how many characters have been typed, but not what they are. As the name suggests, jPasswordField is intended to be used as a simple way to accept a user's password. Whereas in case of jTextField whatsoever we type is visible as it is.

147. (a)SIZE (b) COLOR (c) FACE
Font tag is a container tag.

148. Both Loop1 and Loop2 will execute 3 times.
Loop1 and Loop2 both are entry controlled.

149. jTextField1 will show the value of sum As **5**
jTextField2 will show the value of C As **11**

150. The <TR> tag is used to define a table row. A row is made of either table header or table cells. Table rows are not allowed to directly contain data. Instead, table rows act as container for the cells <TD> or <TH> which contain data that is displayed in the table.

151. (i) Just Another Day
(ii) 31

152.
```
if(code==0)
Remarks = "100% Tax Exemption";
else if(code==1)
Remarks = "50% Tax Exemption";
else if(code==2)
Remarks = "30% Tax Exemption";
else
```

Remarks = “Invalid Entry”;

153. jTextField1.setEditable(false);
jTextField1.getText();

154. jTextField1 will be 9 and jTextField2 will be 6

155. Value of Str1 will be HelloDear Friend
Value of Str2 will be Hello

156.

(i)
```
float total;
float eng=Float.parseFloat(jTextField1.getText());
float as=Float.parseFloat(jTextField2.getText());
float gk=Float.parseFloat(jTextField3.getText());
tot= eng+as+gk;
jTextField4.setText(“”+tot);
```

(ii)
```
float total; Char Grade= ‘ ’ ;
total= Float.parseFloat(jTextField4.getText());
float tot=total/3;
if(tot>=80)
    Grade= ‘A’;
else if(tot>65)
    Grade= ‘B’;
else if(tot>50)
    Grade=  ‘C’;
else
    Grade=’D’;
jTextField5.setText(“”+Grade);
```

(iii) System.exit(0);

157. No. Examples are : “A”, “India”

158. String City= “Ambala”;

159. ‘/’ divides first number with second number and returns the quotient.
‘%’ divides first number with second number and returns the remainder.
Example 7/2 will return 3 and 7%2 will return 1

160. (i) <IMG> (ii) <P> or

161. agg = 9 agg1 = 9

162. 25 16 9 4

163. ALIGN , BGCOLOR , BORDER

164. Object Oriented Programming (OOP) is a paradigm that lays emphasis on data. It represents instance of a class as objects. It has data members and associated methods. Languages such as C++, Java supports OOPs.

165.
```
int option = Integer.parseInt(jTextField1.getText());
switch(option)
{
case 1: jTextField2.setText("Regular Employee"); break;
case 2: jTextField2.setText("On Probation"); break;
case 3: jTextField2.setText("Visiting Faculty"); break;
case 4: jTextField2.setText("On Contract"); break;
default: jTextField2.setText("Invalid Option");
}
```

166. PEACESpread

167.
```
jTextField1.setEnabled(false);
jTextField1.setText("Welcome");
        OR
jTextField1.disable();
jTextField1.setText("Welcome");
```

168. 5

169.
```
jTextField2 = 15
jTextField3 = 65
```

170. (i)
```
int wfee=0,tfee=0;
int nop= Integer.parseInt(jTextField1.getText());
int efee= nop * 500;
if(jCheckBox1.isSelewcted())
        wfee = wfee * 250;
tfee = efee + wfee;
jTextField2.setText(""+ efee);
jTextField3.setText(""+ wfee);
```

jTextField4.setText(""+ tfee);

(ii) jTextField1.setText(" ");
jTextField2.setText(" ");
jTextField3.setText(" ");
jTextField4.setText(" ");
OR
jTextField1.setText(NULL);
jTextField2.setText(NULL);
jTextField3.setText(NULL);
jTextField4.setText(NULL);

(iii) System.exit(0);

171.

if-else	**switch-case**
If can evaluate a relational or logical expression i.e., multiple conditions.	Switch can only test for equality.
The if-else constructions lets you use a series of expressions that may involve unrelated variables and complex expression.	The switch statement selects its branches by testing the value of same variable.
The if-else statement can handle floating point test also apart from handling integer and character tests.	A switch cannot handle floating point tests. Thecase labels of switch must be non fractional.

172.

while	**do-while**
while is an entry-controlled loop.	do-while is a exit-controlled loop.
In while loop the test expression is evaluated at the begining of the loop i.e., before executing the loop body.	In do-while loop the test expression is evaluated at the end of the loop i.e., after executing the loop body.

Syntax:
while(test-expression)
{
// statement
}
	Syntax:
	do
	{
	// statement
	} while(test-expression);

while loop will not execute even once if text expression is false.	do while loop will execute atleast once although text expression is false.

173. Two purposes of "+"
(i) "+" is used to add to numbers.
(ii) "+" is used to concatenate two strings.

174. X
X

175. The test condition is always false

176. Second String
Third String

177. 4 5 6 6 7
The number 6 is printed twice

178. We can use colspan and rowspan attributes of the <TD> tag.

179.

Container Element/tag	Empty Element/tag
This type of element requires starting as well as matching closing tag	This type of element requires only starting tag
These tags affect everything that comes between their starting and ending tag.	Empty tag is just carry out their specific job
Eg. <Font> …. </Font> <B> ….. </B>	Eg. <HR>

180. He can use float or double type of Variable. double num;

float num;

181. The value of both num and num1 will be 102.

182. She should use the selectionMode property. Selection modes are SINGLE_SELECTION, SINGLE_INTERVAL_SELECTION, MULTIPLE_INTERVAL_SELECTION

183. "break" statement is missing after every case.

184. (i) Import the packages required for database programming.
(ii) Register the JDBC Driver.
(iii) Open a connection.
(iv) Execute a query.
(v) Extract data from result set.
(vi) Clean up the environment.

185. **ByteCode:** A bytecode is a byte-long instruction that the Java compiler generates and the Java interpreter executes.
JVM: JVM(Java Virtual Machine) is a program which behaves as interpreter and translate bytecode into machine languages as they go- called just-in-time compilation(JIT).

186. Each instance variable is initialized with a default value when it is created:

Type	Initial / Default value	Type	Initial / Default value
byte	0(Zero of byte type)	double	0.0D
short	0(Zero of byte type)	char	null character i.e., '\u000'
int	0	boolean	false
long	0L	All reference types	null
float	0.0F		

187. The import statement in Java allows programmers to refer to classes which are declared in other packages to be accessed without referring to the full package name. Import statement must be the first line of the program. Import statement have 2 option for importing the classes from packages –

a. single class from package by mentioning the class name specifically.
   Eg. Import javax.swing.jOptionPane;

b. entire class by using * (asterisk) symbol in place of class name.
   Eg. import javax.swing.*;

Classes from the **java.lang** package are automatically imported, so it is not require to explicitly importing.

188. (i) Import the packages required for database programming.
(ii) Register the JDBC Driver.
(iii) Open a connection.
(iv) Execute a query.
(v) Extract data from result set.
(vi) Clean up the environment.

189.

Call By Value	Call by reference
Call by value is used to create a temporary copy of the data which is transferred from the actual parameters in the formal parameters.	Call by reference is used to share the same memory location for actual and formal parameters.
The changes in the formal parameters are not reflected back in the actual parameters.	The changes in the formal parameters are reflected back in the actual parameters.

190. Containers are the controls that can hold other controls within it e.g., a Frame(there can be multiple controls inside a frame) or a label(it can hold an image and/or text) or jPanel (we can put so many controls in it). Controls inside a container are known as child controls. The child controls can exist completely inside their containers. That means we can's move them outside

their container.

When we delete a container control, all its child controls automatically get deleted.

191.

List Box	Combo Box
(i) A list doesn't have a text field	A combobox is a combination of text field and a list box.
(ii) In a list, the user must select items directly from the list.	In a combobox user can selelc an item by typing the text as well as mouse click
(iii)The list doesn't drop down.It consumes more space of jFrame	A combobox takes less space initially but drops down when user clicks on its arrow.
(iv) Lists allow us to select more than one item at a time.	Combobox allows only single item at a time.

192. The lifetime of variable is the period during which it can be accessed.

The scope of a variable is the area of code where the variable can be accessed.

A local variable is declared inside a method and is accessible only inside that method. Scope and Lifetime of a local variable is less as compared to global variable.

193. **Package:** A package is a Java language element used to group related classes under a common name.

java.lang: Provides classes that are fundamental to the design of the Java programming language. The most important classes are Object, which is the root of the class hierarchy, and Class, instances of which represent classes at run time.

java.util: Contains the collections framework, date and time facilities, a random-number generator.

194.
```
double cal_power(int x, int y)
  {
  double z=Math.pow(x,y);
  return(z);
  }
```

195. (i) **getString() :** getString() method is provided by ResultSet object and used for obtaining column data for a row.

Example: String Tname=rs.getString("Name"); // Here **Name** is a column the table

(ii) **substring():** this method is used to return a part of the String .

The first argument represents the starting index of the string and second (optional) argument represents the stoping index of the string

Example: String s="Welcome";
System.out.println(s.substring(3)); // Output will be **come**
System.out.println(s.substring(3,5)); // Output will be **co**

196. (i) **getSelectedValue():** Returns the selected value when only a single item is selected in the list. When multiple items are selected, it returns the first selected value. Returns null if there is no selection.

Example: String player=(String) jList1.getSelectedValue();
jLabel1.setText(player);

(i) getSelectedItem(): Returns the selected item from the combo box
Example: String job =(String) jComboBox1.getSelectedItem();
jLabel1.setText(job);

197.
```
public void fun()
    {
      int a=Integer.parseInt(jTextField1.getText());
      int b=Integer.parseInt(jTextField2.getText());
       int sum=a+b;
        JOptionPane.showMessageDialog(this,sum);
    }
```

198.

(i)
```
txtDiscount.setEditable(false);          // txtDiscount is name of jTextField
txtNetAmt.setEditable(false);           // txtNetAmt  is name of jTextField
```

(ii)
(a) //Code for calculate Discount button

```
String name= txtname.getText();
double bill=Double.parseDouble(txtbillamt.getText(););
double disc=0.0;

String s= cmbMode.getSelectedItem().toString();
if(s.equals("Cash"))
{
        disc= 0.08*bill;
}
else if(s.equals("Cheque"))
{
        disc=0.07*bill;
}
else if(s.equals("Cash"))
{
disc=0;
```

```
}
txtDiscount.setText(" "+disc);

    (b)  //Code for calculate Net Amount button
      double  netAmt=0.0
     double bill=Double.parseDouble(txtbillamt.getText());
     double  disc= Double.parseDouble (txtDiscount.getText());

      netAmt=bill-disc;
      txtNetAmt.setText(" "+netAmt);

(iii)
      System.exit(0);
```

```
199. Ans. private void jButton1ActionPerformed (java.awt.ActionEvent evt)
        {
                double sales_amt = 0.0;

                if (! txtSales.getText( ).trim( ).equals( ""))
                {
                        sales_amt=Double.parseDouble(txtSales.getText( ));
                }

                double incentive = 0.0;
                if (jCheckBox1.isSelected ( ))
                {
                        incentive = incentive + 0.1;
                }
                if (jCheckBox2.isSelected ( ))
                {
                        incentive = incentive + 0.8;
                }
                if (jCheckBox3.isSelected ( ))
                {
                        incentive = incentive + 0.05;
                }

                double total_incentive= sales_amt * incentive;
                txtIncentive.setText ( " " + total_incentive);
}
```

```
200.  (i)   // coding for Calculate Button

        if(jchkStrawberry.isSelected()==true)
                jTxtPriceStrawberry.setText("45");
        else
        {
                jTxtPriceStrawberry.setText("0");
```

```
                jTxtQtyStrawberry.setText("0");
        }
        if(jChkChocolate.isSelected()==true)
                jTxtPriceChocolate.setText("60");
        else
        {
                jTxtPriceChocolate.setText("0");
                jTxtQtyChocolate.setText("0");
        }
        if(jChkVinella.isSelected()==true)
                jtxtPriceVinella.setText("40");
        else
        {
                jtxtPriceVinella.setText("0");
                jTxtQtyVinella.setText("0");
        }
        int r1,r2,r3,q1,q2,q3,a1,a2,a3,gt;
        r1=Integer.parseInt(jTxtPriceStrawberry.getText());
        r2=Integer.parseInt(jTxtPriceChocolate.getText());
        r3=Integer.parseInt(jtxtPriceVinella.getText());
        q1=Integer.parseInt(jTxtQtyStrawberry.getText());
        q2=Integer.parseInt(jTxtQtyChocolate.getText());
        q3=Integer.parseInt(jTxtQtyVinella.getText());
        a1=r1*q1;
        jTxtAmtStrawberry.setText(""+a1);
        a2=r2*q2;
        jTxtAmtChocolate.setText(""+a2);
        a3=r3*q3;
        jTxtAmtVinella.setText(""+a3);
        gt=a1+a2+a3;
        jTxtTotalAmt.setText(""+gt);

(ii)    // coding for Clear Button

          jTxtPriceStrawberry.setText("");
        jTxtPriceChocolate.setText("");
        jtxtPriceVinella.setText("");
        jTxtQtyStrawberry.setText("");
        jTxtQtyChocolate.setText("");
        jTxtQtyVinella.setText("");
        jTxtAmtStrawberry.setText("");
        jTxtAmtChocolate.setText("");
        jTxtAmtVinella.setText("");
        jchkStrawberry.setSelected(false);
        jChkChocolate.setSelected(false);
        jChkVinella.setSelected(false);

(iii)   // coding for Close Button
          System.exit(0);
```

201. (i)
```
double bs=0,da=0,net=0,ded=0,gross=0,hra=0;
if (rdnon.isSelected()==true)
{
 bs=10000;
 da=(31*bs)/100;
 hra=(30*bs)/100;
 ded=(12*bs)/100;
}
else if (rdpgt.isSelected()==true)
{
 bs=20000;
 da=(30*bs)/100;
 hra=(30*bs)/100;
 ded=(12*bs)/100;
}
else if (rdtgt.isSelected()==true)
{
 bs=15000;
 da=(21*bs)/100;
 hra=(30*bs)/100;
 ded=(12*bs)/100;
}
else if (rdprt.isSelected()==true)
{
 bs=12500;
 da=(20*bs)/100;
 hra=(25*bs)/100;
 ded=(12*bs)/100;
}
gross=bs+da+hra;
net = gross – ded;

if(chk10.isSelected()==true)
{
 net=net+3000;
}

tfbs.setText(“ ”+bs);
tfded.setText(“ ”+ded);
tfgross.setText(“ ”+gross);
tfnet.setText(“ ”+net);
```

(ii)
```
System.exit(0);
```

(iii)
```
tfgross.setEditable(false);
 tfded.setEditable(false);
 tfnet.setEditable(false);
```

202. (i)
```
int p;
p=Integer.parseInt(jTextField2.getText());
if (jCheckBox1.isSelected())
p=p+5;
jTextField3.setText(Integer.toString(p));
```

(ii)
```
int p;
p=Integer.parseInt(jTextField3.getText());
if( jRadioButton1.isSelected())
{
        if ( p>=70)
                jTextField4.setText("Eligible for all subject");
        else
                jTextfield4.setText("Not Eligible for science");
}
else if( jRadioButton2.isSelected())
{
        if ( p>=60 )
                jTextField4.setText("Eligible for Commerce and Humanities");
        else
                jTextfield4.setText("Not Eligible for Science and Commerce");
}
else
{
        if ( p>=40 )
                jTextField4.setText("Eligible for Humanities");
        else
                jTextfield4.setText("Not Eligible for any subject ");
}
```

(iii)
```
jTextField1.setText(" ");
jTextField1.setText(" ") ;
jTextField1.setText(" ") ;
jTextField1.setText(" ") ;
jCheckbox1.setSelected(false);
```

OR

```
jTextField1.setText(null);
jTextField1.setText(null);
jTextField1.setText(null);
jTextField1.setText(null);
jCheckbox1.setSelected(false);
```

(iv)
```
System.exit(0);
```

203. x will be 7 and y will be 4

204. In jTextField1 value **2** will display
In jTextField2 value **8** will display

205. (i) Button Group control should be used.
(ii)

(a)
```
Int wage_rate=0;
if(rbMale.isSelected())
wage_rate= 140;
else if(rbFeMale.isSelected())
wage_rate= 160;

if(chkSkilled.isSelected())
wage_rate= wage_rate+50;

int  days= Integer.parseInt(txtDays.getText());
int total_wages= wage_rate*days;
lblWages.setText(“”+total_wages);
```

(b)
```
txtName.setText(“”);
txtDays.setText(“”);
```

(c)
```
System.exit(0);
```

206.
```
int fun(int N)
{
  int N=N/10;
  int D= N % 10;
  return D;
}
```

207.
```
public int fun(int a,int b)
{
  if(a*a >= b*b)
   return a;
  else
   return b;
}
```

208.

HTML	XML
It is presentation oriented not data oriented	It supports data representation through standard

	data structure
It does not have data validation capabilities	It has data validation capabilities
With it data cannot be shared across applications	With it same data can be viewed in multiple ways by different user groups and applications

209. (i) 131 (ii) AB (iii) AB (iv) AB (v) 131

210. 1238

211. (i) // coding for calcBTN

```
int mks= Integer.parseInt(marksTf.getText());
String  grade = "You get";
if(mks>=90)
   grade=grade+ "A++";
else if((mks>=80)
   grade=grade+ "A+";
else if((mks>=75)
   grade=grade+ "A";
else if((mks>=60)
   grade=grade+ "B";
else if((mks>=50)
   grade=grade+ "C";
else if((mks>=40)
   grade=grade+ "D";
else
   grade= "You Failed";
resultLBL.setText(grade);
```

(ii)

```
private void marksTFFocusLost(.....)
{
    int mks= Integer.parseInt(marksTf.getText());
    if(mks<=0)
    {
        JOptioPane.showMessageDialog(null,"Marks cannot be negative or zero");
       marksTF.setText("");
       marksTF.requestFocus();
    }
    }
```

(iii)

```
marksTF.setText("");
resultLBL.setText("");
```

212. 6 6

213. **Constructor function** is a the function which has

(i) same name as its class is called constructor

(ii) it is no return type, not even void

(iii) used to initialize the objects of that class type

(iv) Automatically executes when we create the object of the class

By using **super** keyword we can access super class constructor from sub class.

214.
```
public int fun(int CP,int SP)
        {
    int profit=0;
        if(SP>CP)
       profit=SP-CP;
       return profit;
  }
```

215. Loop will execute Infinite times.

216.
```
i=2;
        while(i<51)
         {
           JOptionPane.showMessageDialog(null,i+ "");
            I+=2;
        }
        JOptionPane.showMessageDialog(null,"Thank You");
```

217.
hello world
Hello!world

218.
NFOR
INFORMATICS PRACTICES

219. The *true*, *false*, and *null* are not Java *keywords*, but they are *reserved words*. Remember that all keywords are reserved words, but not vice versa. Java keywords and reserved words cannot be used as identifiers such as variable, class, or method names.

220.
```
public String fun(int num)
 {
   int d,h;
  String output= "";
   h= num/2;
   for(d=2;d<=h;d++)
   {
     if(num%d==0)
      break;
    }
if(d>h)
  output= "Yes it is prime";
else
output= "Not Prime";
return output;
}
```

221.
```
import javax.swing.JOptionPane;
public boolean fun()
{
 String A=JOptionPane.showInputDialog("Enter a Number");
 int num=Integer.parseInt(A);
 int num1=num;
  int D,S=0;
  while(num>0)
   {
     D=num%10;
      S=S+ D*D*D;
      num=num/10;
    }
  if(S==num1)
   return true;
  else
     return false;
}
```

222.
```
 int num=Integer.parseInt(jTextField1.getText());
   int D,rev=0;
  while(num>0)
   {
     D=num%10;
     rev=rev*10 +D;
      num=num/10;
    }
  jLabel1.setText(""+rev);
```