**Guess Paper – 2013
Class – XII
Subject – Informatics Practices**

 **IP RECORD**

1.greatest of three numbers .



Sol. private void grButtonActionPerformed(java.awt.event.ActionEvent evt) {

 float N1, N2, N3; // Variables to hold three input values.

 float max; // Variable to hold maximum value.

 N1 = Integer.parseInt(txtN1.getText());

 N2 = Integer.parseInt(txtN2.getText());

 N3 = Integer.parseInt(txtN3.getText());

 // only checks the nonzero value

 if ((N1 >= 0) && (N2 >= 0) && (N3 >= 0)) {

 max = N1;

 if (N2>max)

 max = N2;

 if (N3>max)

 max = N3;

 jLabel4.setText("The greater number is : " +max);

 }

}

 private void exButtonActionPerformed(java.awt.event.ActionEvent evt) {

 System.exit(0);

}

**2.moving data from text fields to text area**

 

Sol. private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

 jTextArea1.append("Friend's Data" + "\n"); // Roll No.

 jTextArea1.append("Roll No.: " + jTextField1.getText() + "\n"); // Roll No.

 jTextArea1.append("Name: " + jTextField2.getText() + "\n"); // Name

 jTextArea1.append("Address: " + jTextField3.getText() + "\n"); // Address

 jTextArea1.append("Section: " + jTextField4.getText() + "\n"); // Section

 jTextArea1.append("Grade: " + jTextField5.getText() + "\n"); // Grade

}

3. display salary according to designation .

 

Sol. private void btnSalActionPerformed(java.awt.event.ActionEvent evt) {

 String Desig = jComboBox1.getSelectedItem().toString();

 if (Desig.equals("Project Manager")) {

 jTextField1.setText("200000");

 } else if (Desig.equals("Manager")) {

 jTextField1.setText("150000");

 } else if (Desig.equals("Architect")) {

 jTextField1.setText("100000");

 } else if (Desig.equals("Team Leader")) {

 jTextField1.setText("100000");

 } else if (Desig.equals("Sr. Programmer")) {

 jTextField1.setText("75000");

 } else if (Desig.equals("Programmer")) {

 jTextField1.setText("50000");

 } else if (Desig.equals("Operator")) {

 jTextField1.setText("25000");

 }

}

 private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {

 System.exit(0);

}

4. arthimetic operations

 

Sol. private void btnAOActionPerformed(java.awt.event.ActionEvent evt) {

 int n1, n2, aoPlus, aoMult, aoRem, Diff;

 float aoDiv;

 n1 = Integer.parseInt(txtNum1.getText());

 n2 = Integer.parseInt(txtNum2.getText());

 aoPlus = n1 + n2 ;

 Diff = n1 - n2;

 aoMult = n1 \* n2;

 aoDiv = n1 / n2;

 aoRem = n1 % n2;

 txtPlus.setText(String.valueOf(aoPlus));

 txtMinus.setText(String.valueOf(Diff));

 txtMult.setText(String.valueOf(aoMult));

 txtDiv.setText(String.valueOf(aoDiv));

 txtMod.setText(String.valueOf(aoRem));

}

 private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {

 System.exit(0);

}

5. find the reverse of a given no.

 

Sol. private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

 System.exit(0);

}

 private void btnRevActionPerformed(java.awt.event.ActionEvent evt) {

 String str;

 int value, r\_digit;

 value = 0;

 while (value <= 0) {

 value = Integer.parseInt(txtNum.getText());

 if (value <= 0) {

 JOptionPane.showMessageDialog(this, "The number must be positive");

 break;

 }

 }

 str = " ";

 while (value != 0) {

 r\_digit = value % 10;

 str = str + Integer.toString(r\_digit);

 value = value / 10;

 }

 txtRev.setText(str);

}

 6. calculate percentage and grade



Sol. private void cmdCalcGradeActionPerformed(java.awt.event.ActionEvent evt) {

 float per = Float.valueOf(txtPer.getText());

 String gr="";

 // Medical section

 if (optMed.isSelected()) {

 if (per >= 80)

 gr = "A";

 else

 if (per >= 60 && per <= 78)

 gr = "B";

 else

 if (per < 60)

 gr = "C";

 }

 // Non-Medical section

 else

 if (optNMed.isSelected()) {

 if (per >= 75)

 gr = "A";

 else

 if (per >= 50 && per <= 74)

 gr = "B";

 else

 if (per < 50)

 gr = "C";

 }

 txtGr.setText(gr);

}

 private void cmdClearActionPerformed(java.awt.event.ActionEvent evt) {

 txtFirst.setText("");

 txtSecond.setText("");

 txtPer.setText("");

 txtGr.setText("");

 chkNCC.setSelected(false);

 optMed.setSelected(true); // Default button selected

 optNMed.setSelected(false);

}

 private void cmdExitActionPerformed(java.awt.event.ActionEvent evt) {

 System.exit(0);

}

 private void btnDisActionPerformed(java.awt.event.ActionEvent evt) {

 txtPer.enable(false);

 txtGr.enable(false);

 optMed.setSelected(true); // Default button selected

 optNMed.setSelected(false);

}

 private void cmdCalcPerActionPerformed(java.awt.event.ActionEvent evt) {

 int fTerm, sTerm;

 int Total=0;

 float per=0;

 fTerm = Integer.parseInt(txtFirst.getText());

 sTerm = Integer.parseInt(txtSecond.getText());

 Total = fTerm + sTerm; // Total marks

 per = Total/2; // Percentage of marks

 if (chkNCC.isSelected())

 per = per + 3; // Extra 3% is given as NCC Cadet

 // Displaying percentage

 txtPer.setText(Float.toString(per));

}

7.class report

 

Sol. class Report

 {

 int adno;

 String Name;

 float M1, M2, M3, M4, M5;

 float average;

 float getavg( )

 {

 return ((M1 + M2 + M3 + M4 + M5)/5);

 }

 void read\_info( )

 {

 adno = Integer.parseInt(txtAdm.getText());

 Name = txtSname.getText();

 M1 = Integer.parseInt(txtM1.getText());

 M2 = Integer.parseInt(txtM2.getText());

 M3 = Integer.parseInt(txtM3.getText());

 M4 = Integer.parseInt(txtM4.getText());

 M5 = Integer.parseInt(txtM5.getText());

 average = getavg();

 }

 void displayinfo()

 {

 txtAvg.setText(Float.toString(average));

 if (average > 40) {

 jLabel5.setText("You are passed");

 jLabel5.setForeground(Color.blue);

 }

 else {

 jLabel5.setText("You are failed");

 jLabel5.setForeground(Color.red);

 }

 }

 }

 private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {

 System.exit(0);

}

 private void btnAvgActionPerformed(java.awt.event.ActionEvent evt) {

 // Report class object

 Report RP = new Report();

 // Member method accessed through class object

 RP.read\_info();

 RP.displayinfo();

}

8.create an application using list and combo boxes

 

Sol. private void cmdListActionPerformed(java.awt.event.ActionEvent evt) {

 String City = txtCity.getText();

 DefaultListModel dModel = (DefaultListModel) jList1.getModel();

 dModel.addElement(City);

 jList1.setModel(dModel);

}

 private void cmdLCActionPerformed(java.awt.event.ActionEvent evt) {

 DefaultListModel dModel = (DefaultListModel) jList1.getModel();

 dModel.clear();

}

 private void cmdComboActionPerformed(java.awt.event.ActionEvent evt) {

 String City = txtCity.getText();

 DefaultComboBoxModel cModel = (DefaultComboBoxModel) jComboBox1.getModel();

 cModel.addElement(City);

 jComboBox1.setModel(cModel);

}

 private void cmdCCActionPerformed(java.awt.event.ActionEvent evt) {

 DefaultComboBoxModel cModel = (DefaultComboBoxModel) jComboBox1.getModel();

 cModel.removeAllElements();

}

 private void cmdLCIndActionPerformed(java.awt.event.ActionEvent evt) {

 String City = (String) jList1.getSelectedValue();

 int ind = jList1.getSelectedIndex();

 if (jList1.isSelectedIndex(ind)) {

 DefaultListModel dModel = (DefaultListModel) jList1.getModel();

 dModel.remove(ind);

 JOptionPane.showMessageDialog(this, "Deleted name " + City);

 jList1.setModel(dModel);

 } else

 JOptionPane.showMessageDialog(this, "Note name is selected from list");

}

 private void cmdClearActionPerformed(java.awt.event.ActionEvent evt) {

 txtCity.setText("");

}

 private void cmdCBIndActionPerformed(java.awt.event.ActionEvent evt) {

 String City = (String) jComboBox1.getSelectedItem();

 int ind = jComboBox1.getSelectedIndex();

 DefaultComboBoxModel cModel = (DefaultComboBoxModel) jComboBox1.getModel();

 cModel.removeElementAt(ind);

 JOptionPane.showMessageDialog(this, "Deleted name " + City);

 jComboBox1.setModel(cModel);

9. String manipulation



Sol. private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

 System.exit(0);

}

 private void btnStringActionPerformed(java.awt.event.ActionEvent evt) {

 String Name = txtName.getText();

 String School = txtSch.getText();

 String nName = Name.concat(School); // Concatenate Name with School

 int nlength = Name.length();

 String nStr = Name.substring(9); // Index stats from 9th position

 String nStr1 = School.substring(9, 13); // Index start from 9th position till 13th

 String uCase = Name.toUpperCase(); // Converts into uppercase letters

 String LCase = School.toLowerCase(); // Converts into lowercase letters

 String mess1 = " My Personal Bio-Data ";

 String Year = "2010";

 String nTrim = mess1.trim() + " " + Year;

 txtStringArea.append("Concatenated string: " + nName + "\n");

 txtStringArea.append("Length of '" + Name + "' is: " + nlength + "\n");

 txtStringArea.append("Name.substring(9) is: " + nStr + "\n");

 txtStringArea.append("School.substring(9, 13) is: " + nStr1 + "\n");

 txtStringArea.append("Name.toUpperCase() is: " + uCase + "\n");

 txtStringArea.append("School.toLowerCase() is: " + LCase + "\n");

 txtStringArea.append("mess1 trim is: " + nTrim + "\n");

}

10. create a desktop application

 

Sol. private void btnIncActionPerformed(java.awt.event.ActionEvent evt) {

 int sales = 0;

 if (!txtSales.getText().trim().equals("")) {

 sales = Integer.parseInt(txtSales.getText().trim());

 }

 double incentive = 0.0;

 if (jCheckBox1.isSelected()) {

 incentive = incentive + 0.1; // 10%

 }

 if (jCheckBox2.isSelected()) {

 incentive = incentive + 0.08; // 8%

 }

 if (jCheckBox3.isSelected()) {

 incentive = incentive + 0.05; // 5%

 }

 txtInc.setText("" + Math.round(sales \* incentive));

}