basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

TRIAL EXAMINATION

INFORMATION TECHNOLOGY P1

2015

MEMORANDUM

MARKS: 150

This memorandum consists of 18 pages.

GENERAL INFORMATION:

- These marking guidelines are to be used as the basis for the marking session. They were prepared for use by markers. All markers are required to attend a rigorous standardisation meeting to ensure that the guidelines are consistently interpreted and applied in the marking of candidates' work..

- Note that learners who provide an alternate correct solution to that given as example of a solution in the marking guidelines will be given full credit for the relevant solution, unless the specific instructions in the paper was not followed or the requirements of the question was not met

- Pages 3-7 include the marking grid for each question for using either one of the two programming languages.

- **P**ages 8-18 contain examples of solutions for Java for Questions 1 to 3 in programming code.

- Copies of Pages 3-7 should be made for each learner and completed during the marking session.

ANNEXURE A:

SECTION A:

QUESTION 1: MARKING GRID- GENERAL PROGRAMMING SKILLS

CENTRE NUMBER:	**EXAMINATION NUMBER:**		
QUESTION	**DESCRIPTION**	**MAX. MARKS**	**LEARNER'S MARKS**
1.1	**Button - [Question 1.1]** Extract Type from the combo box (to be used as text) ✓ Extract Name from the text field. ✓ Construct a line of text using the Name and Type and assign the string to the label provided✓	**3**	
1.2	**Button - [Question 1.2]** Extract the id or vat registration number from the text box✓ Assign a string variable to an empty string. ✓ Assign an integer variable to zero✓ Randomly generate a three digit number✓ Get the first character from the name✓ Get the last two characters from the name✓ Loop through the ID/vat registration number. ✓ Get the character at the loop position and convert to integer and add the digits. ✓ Join the random number to the first and last two characters and the sum to get the account number. ✓ Place the account number in the text box. ✓	**10**	
1.3	**Button - [Question 1.3]** Get the number of smartphones from the spinner. ✓ Get the number of laptops from the spinner. ✓ Get the number of tablets from the spinner. ✓ Calculate the cost for smartphones using the nested if or switch statement. ✓ ✓ ✓ ✓ ✓ Calculate the cost for laptops ✓and tablets and adding to the cost of smartphones. ✓ Calculate the vat and add to the cost. ✓ Format and display the cost in the text box✓	**12**	

1.4	**Button - [Question 1.4]** Input the number of months ✓ Validate the month to be 12 or 24✓ Multiply the interest rate of 5% to the number of years(number of months divided by 12) ✓ Add 1 to the value in step 2✓ Multiply the answer in step 3 to the total cost ✓ Divide the answer in step 4 by the number of months. ✓ Format the answer in step 5 to two decimal places✓ Display the formatted answer in the text area. ✓	**8**	
1.5	**Button - [Question 1.5]** Enable the checkbox, text field and radio buttons in the event handler of the combo box when deliver is selected. ✓✓✓ Get the selected item when the combo box is selected✓ If deliver is selected✓ Get the weight of the items✓ If the checkbox is selected add R2000 to the cost for insurance✓ If Express radio button is selected Set the cost to weight * 50✓ If Standard radio button is selected Set the cost to weight * 30✓ Display the heading for the delivery charges✓ Display the delivery charges formatted to two decimal places. ✓ If items are picked up Calculate the discount 0.05 * total cost✓ Decrease the Total cost by the discount✓ Display the discount formatted to two decimal places✓ Display the Final Cost formatted to two decimal places✓	**15**	
	TOTAL:	**48**	

ANNEXURE B:
SECTION B:

QUESTION 2: MARKING GRID - OBJECT-ORIENTED PROGRAMMING

CENTRE NUMBER:	**EXAMINATION NUMBER:**		
QUESTION	**DESCRIPTION**	**MAX. MARKS**	**LEARNER'S MARKS**
2.1.1	**Constructor:** All parameters ✓ Correct data types for parameters ✓ Assign parameter values to attributes✓	**3**	
2.1.2	**overDueMethod method** Return type correct✓ Get current date from System's clock✓ Convert the date to string ✓ If amoutDue >0 AND ✓currentDate > amountDueDate✓ return true✓	**6**	
2.1.3	**calcInterest method:** Return type correct✓ Check if overDue✓ interest = 12/100.0 * amountDue✓ [learners would have worked out interest per month – mark accordingly] Else interest = 0✓	**4**	
2.1.4	**updateDataBalance method:** Receive TWO parameters - correct data type ✓ Convert gigabytes to megabytes ✓ Increase the dataBalance by the correct parameter✓ Increase the amount due by the correct parameter✓	**4**	
2.1.5	**toString METHOD:** Format and correct attributes✓✓✓ (-1 for each incorrect attribute) Name: <Name of customer> Cell phone Number : <cellNumber> Data balance : <dataBalance > Due date : <dueDate> Amount due : <amountDue> Format Currency✓ New line ✓ Return correct type✓	**6**	

2.2.1	Create Customer object ✓	**1**	
2.2.2	**Button – [Quest2.2.2]:** Obtain cellphone number from combo box✓ Validate the cellphone number✓ if cellphone not valid clear fields✓ Display a suitable message if invalid ✓ If file does not exist/catch & display message and exit✓ else ✓ Create an object to read from the file✓ While loop through✓ file AND cellphone not found ✓ Read one line from text file✓ Get single, name,id no,cellphone number,data balance,total amount due, and due date info from line of text ✓✓ (Split/pos/Scanner) correct data type✓ Test if cellphone from line matches selected cellphone✓ Instantiate object ✓with all arguments ✓ in the correct order✓ Details displayed using toString method ✓ Enable the Question 2.2.3 buttons✓ If cellphone number does not exist output suitable message✓	**20**	
2.2.3	**Button – [Quest2.2.3]:** Check if overdue✓ Create new text file ✓ using the name field ✓ Write the fields cellnumber, amount due , interest,total amount due✓on separate lines✓ to the text file ✓ Close the text file ✓ Message to indicate file was created ✓ else ✓ Display customers details using toString✓ Display message to indicate that account is not over due✓	**11**	

2.2.3	**Button – [Quest2.2.3]:** Set cost and number of gigabytes to 0 ✓ If one gigabyte button is selected✓ Set cost to 149.00✓ Set gigabyte to 1 If two gigabyte button is selected Set cost to 249.00✓ Set gigabyte to 2 If three gigabyte button is selected Set cost to 299.00✓ Set gigabyte to 3 Call the updateDataBalance using the customer object ✓ Display the customers details using the toString✓	**7**	

	TOTAL:	**62**	

ANNEXURE D: SOLUTION FOR QUESTION 1: JAVA

```
//A possible solution to Question 1

=============================================================================
// Question 1.1
=============================================================================
       String type = cmbType.getSelectedItem()+"";✓
       name = txfName.getText();✓
       lblOutput.setText(name+"----"+type);  ✓

=============================================================================
// Question 1.2
=============================================================================

   int randomNumber = (int)(Math.random()*900 + 100);  ✓
       String code="";✓
       int sum=0;  ✓
       String number = txfNumber.getText();✓
       code = code + name.charAt(0);  ✓
       code = code + name.substring(name.length()-2);  ✓
       for (int k = 0; k < number.length(); k++) {✓
           sum = sum + Integer.parseInt(number.charAt(k)+"");  ✓
       }
       code = randomNumber + code + sum;  ✓
       txfAccNumber.setText(code);  ✓
=============================================================================
// Question 1.3
=============================================================================

       int sp = (int)spinSmartPhones.getValue();✓
       double cost=0;  ✓
       switch(sp)  ✓
       {
           case 1:cost = 3000;
                  break;
     ✓   case 2:cost = 5000;    ✓
                  break;
           case 3:cost = 6000;
                  break;
           default: cost = sp * 1700;  ✓
       }
       int laptops = (int)spinLapTops.getValue();✓
       int tablets = (int)spinTablets.getValue();✓
       cost = cost + (laptops*7200)+(tablets*5500);  ✓✓
       double vat = 0.14 * cost;  ✓
       totalcost = cost + vat;
       txfAns_1_3.setText(df.format(totalcost));  ✓
```

```
//============================================================================
// Question 1.4
//============================================================================
        int months;
        do
        {
            months = Integer.parseInt(JOptionPane.showInputDialog("Enter the
                                number of months"));✓
        }
        while(months!=12 && months!=24 );  ✓
        double step2 = 0.05 * (months/12);  ✓
        double step3 = step2 + 1;  ✓
        double step4 = step3 * totalcost;  ✓
        double step5 = step4/months;  ✓
        String step6 = df.format(step5);  ✓
        txaOutput_1_4.append(months + " monthly installments of :\n");
        txaOutput_1_4.append(step6);  ✓
//============================================================================
// Question 1.5
//============================================================================
private void cmbDeliveryTypeActionPerformed(java.awt.event.ActionEvent evt) {
        if (cmbDeliveryType.getSelectedIndex()==0)  ✓
        {
            chbInsurance.setEnabled(true);
            txfWeight.setEnabled(true);  ✓
            radExpress.setEnabled(true);
            radStandard.setEnabled(true);  ✓
        }
}

private void btnQues1_5ActionPerformed(java.awt.event.ActionEvent evt) {
        int d = cmbDeliveryType.getSelectedIndex();✓
        double cost = 0;
        if (d==0)  ✓
        {
            cost = 250;
            double weight = Double.parseDouble(txfWeight.getText());✓
            if (chbInsurance.isSelected())
            {
                cost = cost + 2000;  ✓
            }
            if (radExpress.isSelected())
            {
                cost = cost + (weight * 50);  ✓
            }
            else
            {
                if(radStandard.isSelected())
                {
                    cost = cost + (weight *30);  ✓
                }
            }
            txaOutput_1_5.append("Delivery charges : \n");✓
```

```
            txaOutput_1_5.append(df.format(cost)+"\n");✓
        }
        else
        {
            if (d==1)
            {
                double discount = 0.05 * totalcost;  ✓
                double finala = totalcost-discount;  ✓
                txaOutput_1_5.append("Discount " + df.format(discount)+"\n");✓
                txaOutput_1_5.append("Final Cost "+df.format(finala)+ "\n");✓
            }

        }
}
```

ANNEXURE E: SOLUTION FOR QUESTION 2: JAVA

```
//A possible solution for the object class
```

OBJECT CLASS: DETAILS

```
package Question2Package;


==========================================================================
// Question 2.1.1
==========================================================================
public Customer(String customerName, String idNumber, String cellNumber, int
dataBalance, double amountDue, String amountDueDate) {
       this.customerName = customerName;
       this.idNumber = idNumber;
       this.cellNumber = cellNumber;
       this.dataBalance = dataBalance;
       this.amountDue = amountDue;
       this.amountDueDate = amountDueDate;
 }√√√
```
(3)

```
==========================================================================
// Question 2.1.2
==========================================================================
    public boolean overDue() {
       boolean flag = false;
       Date now = new Date();√
       String date = sdf.format(now); √
       if (amountDue > 0 && date.compareTo(amountDueDate) > 0) {√√
           flag = true; √
       }
       return flag; √
   }
```
(6)

```
    ==========================================================================
    // Question 2.1.3
    ==========================================================================
   public double calcInterest() {
       double interest = 0; √
       if (overDue()) {√
           interest = 12 / 100.0 * amountDue; √
       }
       return interest; √
   }
```
(4)

```
    ==========================================================================
    // Question 2.1.4
    ==========================================================================

    public void updateDataBalance(int data, double cost) {√
       dataBalance += data*1024; √√
       amountDue += cost; √
   }
```
(4)

```
==========================================================================
// Question 2.1.5
==========================================================================
   public String toString() {
    String  s = "Name: "√ + customerName +"\n"+ ;  √
    s = s +"Cell phone number: " + cellNumber + "\n"; √
    s =s+ "Data Balance: " + dataBalance + " MB" + "\n";
    s=s+ "Due date: " + amountDueDate + "\n";
    s =s+ "Amount due: " + df.format(amountDue) √+ "\n";
    Return s;√
}
```

GUI CLASS: QUESTION2_SOLUTION

```
    ================================================================================
    // Question 2.2.1
    ================================================================================

     Customer obj; √
   ================================================================================
   // Question 2.2.2
   ================================================================================
          private void btnQ2_2_2ActionPerformed(java.awt.event.ActionEvent evt) {
        String cellNo = txfCellNumber.getText();√
        boolean flag = false;
        if ((cellNo.charAt(0) != '0' || cellNo.length() != 10)) {√
            txfCellNumber.setText("");√
            txfCellNumber.requestFocus();
                 JOptionPane.showMessageDialog(null, "Enter a valid cell phone
number");√
        } else {√
            try {
                Scanner scFile = new Scanner(new File("AccountInfo.txt"));√
                while (scFile.hasNext()) {√
                    String line = scFile.nextLine();√
                    Scanner sc = new Scanner(line).useDelimiter("#|;");√
                    String objCustName = sc.next();
                    String idNumber = sc.next();
                    String cellNumber = sc.next();
                    int dataBalance = sc.nextInt();
                    double amountDue = sc.nextDouble();
                    String date = sc.next();√√√
                    if (cellNumber.equals(cellNo)) {√  //if
(line.contains(cellNo))
                        flag = true;
                        objCust = new Customer(objCustName, idNumber, cellNumber,
dataBalance, amountDue, date); √√
                        txaOutput.append(objCust.toString());√
                        btnQ2_2_3.setEnabled(true); √
                        // btnQ2_2_4.setEnabled(true);
                    }

                }
                if (!flag) {
                    JOptionPane.showMessageDialog(null, cellNo + " was not found
in text file");√
                }
            } catch (FileNotFoundException ex) {
                JOptionPane.showMessageDialog(null, "File not found");√
                System.exit(0);
            }
        }
    }
                                                                              (20)
    ================================================================================
   // Question 2.2.3
   ================================================================================

private void btnQ2_2_3ActionPerformed(java.awt.event.ActionEvent evt) {
        if (objCust.overDue())√ {
            try {
                PrintWriter outFile = new PrintWriter(objCust.getCustomerName() +
".txt");√√
```

```
                outFile.println("Cell Number: " + objCust.getCellNumber());
                outFile.println("Amount due: " +
df.format(objCust.getAmountDue()));√
                outFile.println("Interest charged: " +
df.format(objCust.calcInterest()));
                outFile.println("Total amount due: " +
df.format(objCust.getAmountDue() + objCust.calcInterest()));√
                outFile.close();√
                txaOutput.setText(objCust.toString());√
                txaOutput.append("Interest charged: " +
df.format(objCust.calcInterest()) + "\n");
                txaOutput.append("Total amount due: " +
df.format(objCust.getAmountDue() + objCust.calcInterest()) + "\n");√
                txaOutput.append("File has been created\n");√
            } catch (IOException ex) {
            }
        } else {
            txaOutput.setText(objCust.toString());√
            txaOutput.append("Account is not overdue\n");√
            btnQ2_2_4.setEnabled(true);
        }
    }
```

(11)

```
    ============================================================================
    // Question 2.2.4
    ============================================================================
      private void btnQ2_2_4ActionPerformed(java.awt.event.ActionEvent evt) {
        double cost = 0.0;
        int gigs = 0; √
        if (rbtOneGig.isSelected()) {√
            cost = 149.00;
            gigs = 1; √
        } else {
            if (rbtTwoGigs.isSelected()) {
                cost = 249.00;
                gigs = 2; √
            } else {
                if (rbtThreeGigs.isSelected()) {
                    cost = 299.00;
                    gigs = 3; √
                }
            }
        }
        objCust.updateDataBalance(gigs, cost); √
        txaOutput.setText(objCust.toString());√
    }
```

(7)

ANNEXURE F: SOLUTION FOR QUESTION 3: JAVA

```
// A possible solution to Question 3

==========================================================================
    // Question 3.1
    ==========================================================================
String[] arrBrand = new   String[arrSales.length];
    char[] arrCategory = new char[arrSales.length];
    double[] arrPrice = new double[arrSales.length];
    int[] arrQuantity = new int[arrSales.length];
    String[] arrBrandnoDups = new String[arrSales.length]; √√
    int c = 0;

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

        String brand = txfBrandName.getText();√

        txaOutput.setText("Sales for the " + brand + " brand:\n");√
        for (int i = 0; i < arrSales.length; i++) {√

            Scanner sc = new Scanner(arrSales[i]).useDelimiter("#");√
            arrBrand[i] = sc.next();
            arrCategory[i] = sc.next().charAt(0);
            arrPrice[i] = sc.nextDouble();
            arrQuantity[i] = sc.nextInt();√
            System.out.println(arrBrand[i]+"  "+arrQuantity[i]+"  "+arrPrice[i]+"
"+arrCategory[i]); √
            if (arrBrand[i].equalsIgnoreCase(brand)) √ {
                String cat = "Smart Phone";√
                if (arrCategory[i] == 'T') {
                    cat = "Tablet";√
                }

                txaOutput.append(String.format("%-15s%-20s%-10.2f%-5d\n",
arrBrand[i], cat, arrPrice[i], arrQuantity[i])); √
            }
        }                                                              //12
    }
     ==========================================================================
    // Question 3.2
    ==========================================================================
    private void btnViewCategoryActionPerformed(java.awt.event.ActionEvent evt) {
        double totalSmartPhones = 0; √
        double totalTablets = 0; √
        for (int i = 0; i < arrSales.length; i++) {√
            switch (arrCategory[i]) √
            {
                case 'S':{ √
                    totalSmartPhones += arrQuantity[i] * arrPrice[i]; √
                    break;}
                case 'T':{ √
                    totalTablets += arrQuantity[i] * arrPrice[i]; √
                    break;}
            }
         }
        txaOutput.setText("Total sale for smartphones " +
df.format(totalSmartPhones)); √
       txaOutput.append("\nTotal sale for tablets " + df.format(totalTablets)); √√
        btnGetDetails.setVisible(false);
        btnViewCategory.setVisible(false);
        jButton1.setVisible(false);  √
                                                                       //12
         }
```

```
    ============================================================================
    // Question 3.3
    ============================================================================


    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        for (int i = 0; i < arrBrand.length; i++)√  {
            int j = i + 1; √
            boolean flag = true;
            while (j < arrBrand.length && flag) √  {`
                if (arrBrand[i].equalsIgnoreCase(arrBrand[j])) √  {
                    flag = false; √

                }
                j++;√
            }
            if (flag) √  {
                arrBrandnoDups[c] = arrBrand[i]; √
                c++;√
            }
        }
        for (int i = 0; i < c; i++) {
            System.out.println(arrBrandnoDups[i]); √
        }

        int[] arrTotal = new int[c];     //Array is not required
        for (int i = 0; i < c; i++) {√
            for (int j = 0; j < arrBrand.length; j++) {√
                if (arrBrandnoDups[i].equalsIgnoreCase(arrBrand[j])) √  {
                  arrTotal[i]+=arrQuantity[j]; √
                }

            }
      }
        txaOutput.setText(String.format("%-20s%-20s\n", "Brand", "No Sold"));
        for (int i = 0; i < c; i++) {√
            txaOutput.append(String.format("%-20s%-20s\n", arrBrandnoDups[i],
arrTotal[i])); √
        }
                                                                          //16

    }
```