



education

Department:
Education
PROVINCE OF KWAZULU-NATAL

GREENBURY SECVONDARY SCHOOL

INFORMATION TECHNOLOGY P1

GRADE 12

JUNE 2016 EXAMINATIONS

Date of examination: 02 /06/2016

MARKS: 150

EXAMINER(S): D.Chellan
J.Sewnanen
S.Muruvan

TIME: 3 hours

MODERATOR(S): M.Gebreab

This question paper consists of 18 pages.

INSTRUCTIONS AND INFORMATION

1. This question paper is divided into THREE sections. Candidates must answer ALL THREE sections.
2. The duration of this examination is three hours. Because of the nature of this examination it is important to note that you will not be permitted to leave the examination room before the end of the examination session.
3. This question paper is set in programming terms that are not specific to any particular programming language (Java (using the Netbeans IDE)).
4. Make sure that you answer the questions according to the specifications that are given in each question. Marks will be awarded according to the set requirements.
5. Answer only what is asked in each question. For example, if the question does not ask for data validation, then no marks will be awarded for data validation.
6. Your programs must be coded in such a way that they will work with any data and not just the sample data supplied or any data extracts that appear in the question paper.
7. **Routines such as search, sort and selection must be developed from first principles. You may NOT use the built-in features of a programming language for any of these routines.**
8. Data structures which are not supplied must be defined by you, the programmer.
9. You must save your work regularly on the disk/CD/DVD/flash disk you have been given, or on the disk space allocated to you for this examination session.
10. Make sure that your examination number appears as a comment in every program that you code, as well as on every event indicated.
11. If required, print the programming code of all the programs/classes that you completed. You will be given half an hour printing time after the examination session.
12. At the end of this examination session you must hand in a disk/CD/DVD/flash disk with all your work saved on it OR you must make sure that all your work has been saved on the disk space allocated to you for this examination session. Ensure that all files can be read.

13. The files that you need to complete this question paper have been given to you on a disk/CD/DVD/flash disk or on the disk space allocated to you. The files are provided in the form of password-protected executable files.

The following list of files will be available in the folder
IT P1 June2016 DataFiles

Question1:

Question1.form

Question1.java

Question2:

wifilog.txt

Question2GUI.form

Question2GUI.java

WiFiGenerator.java

Question3:

Question3.form

Question3.java

Scenario:

Paradise hotel has four flagship resorts that bring in most of their revenue. The CEO of the hotel wants to ensure smooth running of the hotels and improve services. She has requested 3 programs that will add-value to the business.

SECTION A**QUESTION 1: GENERAL PROGRAMMING SKILLS**

- The project Question1 is provided in the examinations folder
- Open the incomplete class Question1.java.
- Add your name as a comment to the first line of the class

Do the following

- Compile and execute the program. The GUI displays 4 tabs labelled Question 1.1 to Question 1.4. The program currently has no functionality. An example of the GUI is given below
- Complete the code for each section of Question1 as described below

- 1.1 The user is required to enter the surname, firstname and ID number of the guest. A welcome message must then be generated in the format as shown in the sample run below:

SAMPLE RUN

Question 1.1 | Question 1.2 | Question 1.3 | Question 1.4

GUEST SURNAME: Nxumalo

GUEST NAME: Patience

GUEST ID NUMBER: 8402240010083

Welcome to Hilltop Hotel Patience 8402240010083.
We hope your stay with us is memorable.
Please contact reception on 031-2089990
for any assistance

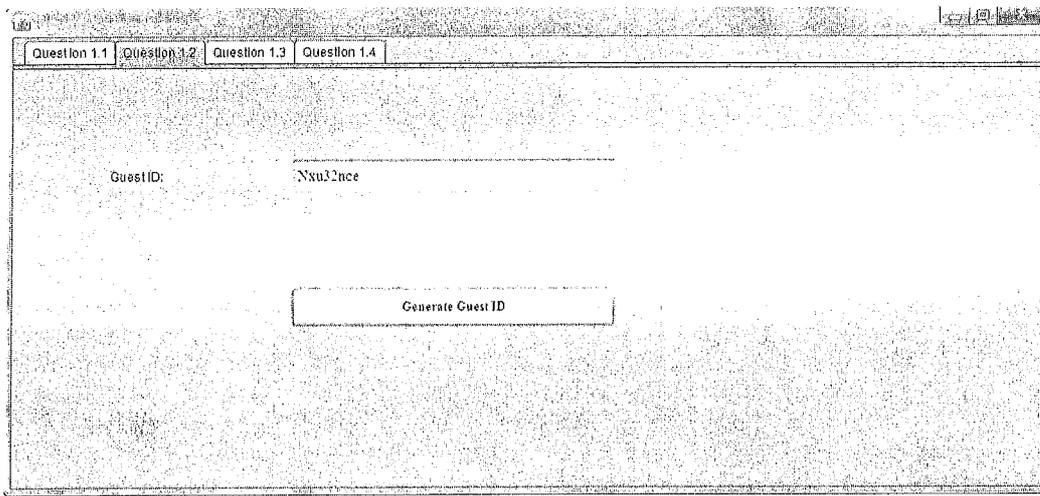
GENERATE WELCOME MESSAGE

(5)

1.2 The program is required to generate a guest ID as follows:

- Concatenate (join) the following characters:
 - The first letter of the guest's surname in upper case
 - The next 2 characters of the guest's surname in lower case
 - The age of the guest
 - The last 3 characters of the guest's name

SAMPLE RUN



(10)

1.3

The guests' can book their accommodation as follows:

TYPE OF ACCOMMODATION	PRICE PER NIGHT
SINGLE ROOM	R1000
FAMILY ROOM	R1500
EXECUTIVE SUITE	R2500

The price of breakfast is R250 per person per day.

Write code to do the following:

- Obtain the type of accommodation from a combo box
- Obtain the number of nights the booking is for
- Determine whether the booking is with or without breakfast
- Determine the total cost according to the user's input
- Payment options:
 - If the guest selects the "CASH" option display the payment in currency format with 2 decimal places in the textbox provided
 - If the user selects the "CARD" option do the following:
 - Use a dialog box to insert the card number.
 - Check the validity of the card number by performing the following test:
 - ✚ The card number must be 10 digits long
 - ✚ The card number must consists of digits only
- If the card is valid add a card charge of 5%. Display the payment in the textbox provided.
- If the card number is invalid do the following:
 - Use a dialog box to display a message that the number is invalid
 - Set the radio button to CASH
 - Display the cash payment

SAMPLE RUN (NEXT PAGE)

1 night stay for 3 guests in a family room with breakfast.

The screenshot shows a web application interface with a navigation bar at the top containing 'Question 1.1', 'Question 1.2', 'Question 1.3', and 'Question 1.4'. The main content area contains a form with the following fields and options:

- Type of accommodation:** FAMILY ROOM (dropdown menu)
- Duration of Stay (No. of nights):** 1 (text input)
- Number of guests:** 3 (text input)
- Payment options:** Cash, Card
- Breakfast:** BREAKFAST
- Buttons:** CALCULATE COST
- Total Payment:** R 2250.00

2 nights stay in a single room with breakfast. Incorrect card number

The screenshot shows the same web application interface as above, but with the following changes:

- Type of accommodation:** SINGLE ROOM (dropdown menu)
- Duration of Stay (No. of nights):** 2 (text input)
- Number of guests:** 2 (text input)
- Payment options:** Cash, Card
- Breakfast:** BREAKFAST
- Buttons:** CALCULATE COST
- Total Payment:** (empty text input)
- Message Box:** A modal message box titled 'Message' is displayed with the text 'Invalid card number' and an 'OK' button.

When ok is clicked. Option is set to cash and payment is shown

The screenshot shows the web application interface after the error message is dismissed. The form is now as follows:

- Type of accommodation:** SINGLE ROOM (dropdown menu)
- Duration of Stay (No. of nights):** 2 (text input)
- Number of guests:** 2 (text input)
- Payment options:** Cash, Card
- Breakfast:** BREAKFAST
- Buttons:** CALCULATE COST
- Total Payment:** R 3000.00

- 1.4 The hotel offers international guests a service that will calculate the conversion of their currency to South African rands. 2 parallel arrays are provided. The array named **arrCountryNames** contains the names of all the popular countries. The array named **arrConversions** contains the current rate of the country's currencies to the South African Rand.

Write code to determine the South African conversion of the currency of a country chosen from a combo box.

SAMPLE RUN

The screenshot shows a web application interface with a navigation bar at the top containing tabs for 'Question 1.1', 'Question 1.2', 'Question 1.3', and 'Question 1.4'. The main content area has three sections: 'CHOOSE YOUR COUNTRY' with a dropdown menu showing 'USA'; 'ENTER AMOUNT YOU WISH TO CONVERT TO S.A. RANDB' with a text input field containing '100'; and 'CONVERTED AMOUNT' with a text input field containing 'R 1497.03'. A 'CONVERT' button is located at the bottom right of the form.

(10)

- Enter your name as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required

TOTAL SECTION A: [50]

SECTION B

QUESTION 2: OBJECT-ORIENTATED PROGRAMMING

SCENARIO

The manager of the hotel requires an app that will produce username(s) and password(s) to allow the guest access to free WiFi for the duration of their stay.

INSTRUCTIONS:

- The project **Question2** is provided and contains an incomplete class called **WiFiGenerator.java** and a GUI form called **Question2GUI.java**.
- Open the incomplete class called **WiFiGenerator.java** in the folder **Source Packages (src), Question2Package**.
- Add your examination number as a comment in the first line of the class called **WiFiGenerator.java**.

Do the following:

- Write code for 2.1.1 to 2.1.6 to make the class functional.

Field Name	Description
hotelName	Hotel name
surname	Guests Surname
roomNumber	A 3-digit room number.
numDays	Number of days that the guest is booked-in.
userName	Comprises of letters and numbers.
password	A code comprising of 4 letters followed by integer values.

- 2.1.1 Use the given information to declare the given **attributes** in the **WiFiGenerator.java** class. (4)
- 2.1.2 Write a **constructor method** that will receive the **hotelName**, the **guests surname**, the **room number** in which the guest will be staying and the **number of days** the guest is staying for as parameters and assigns these values to the attributes. Set the **password** and **userName** attribute to a blank String. (4)
- 2.1.3 Create **accessor** methods for all attributes and a **mutator** method for the **userName** attribute and **password** attribute. (4)
- 2.1.4 Uncomment the given method called **createUserName()** that attaches the **room number** to the end of the **guest surname** and assigns this code to the **username** attribute. (1)

2.1.5 The following algorithm is used to generate a Wifi password:

Step1: set count to 0

Step2: set password to an empty string

Step3: loop 4 times

set pos to a random number from 0 to number of letters in the hotel name.

attach letter at position pos in the hotel name to password.

Step4: multiply the number of nights by 24 and attach this value to the end

of the password.

Step5: set the password attribute in the WiFiGenerator.java class to the generated password

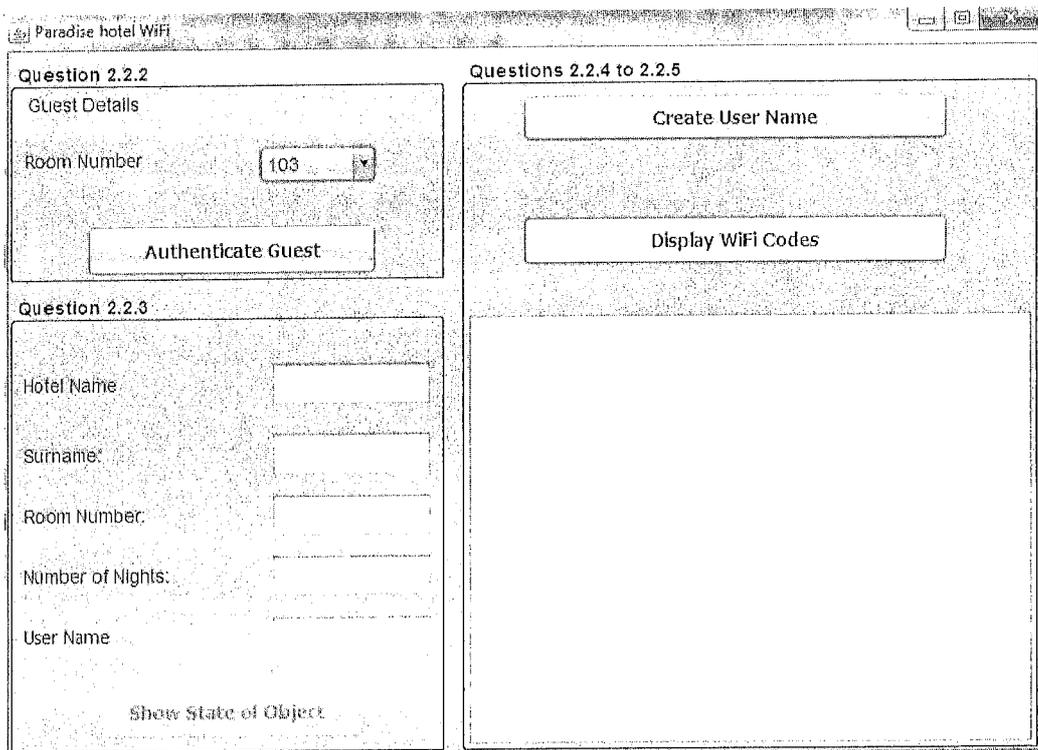
Write a method called **createPassword()** that will create a wifi password using the steps in the algorithm. A while ... loop must be used. Marks will be deducted if the given algorithm is not used.

(8)

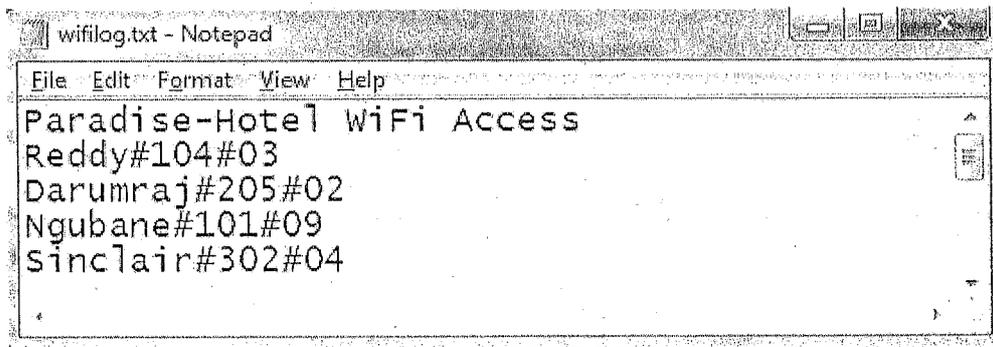
2.1.6 Create a **toString()** method that will return the values of the attributes as indicated in the sample run. (as shown in question 2.2.5)

(5)

2.2 Compile and execute Question2GUI.java in the QuestionTwoPackage.



A text file called **wifilog.txt** has been supplied. The first five lines of the text file is shown below.



```
wifilog.txt - Notepad
File Edit Format View Help
Paradise-Hotel WiFi Access
Reddy#104#03
Darumraj#205#02
Ngubane#101#09
Sinclair#302#04
```

The first line indicates the name of the hotel followed with WiFi Access in the following format:

Name of hotel WiFi Access

Data in the second line onwards is recorded in the following format:

Surname#room number#number of nights

2.2.1 Declare a global object called **guestObj** of the **WiFiGenerator** class. (1)

2.2.2 Write code for the **Authenticate Guest** button that will:

- Obtain the room number from the given combo-box
- Check if the text file wifilog.txt exists. If the file does not exist display the message "File does not exist and terminate the program"
- If the file exists:
 - open the file, read each line and separate the fields
 - if the room number indicated in the combo-box is found instantiate an object
 - if the room number is not found output the message "Room not booked"

Enable the **Show State of Object** button (14)

2.2.3 Write code for the **Show State of Object** button that will write the hotel name, guest surname, room number and number of nights to the corresponding text fields. The user name text field must be left blank.

Paradise Hotel WiFi

Question 2.2.2

Guest Details

Room Number: 103

Authenticate Guest

Question 2.2.3

Hotel Name: Paradise-Hotel

Surname: Maistry

Room Number: 103

Number of Nights: 1

User Name:

Show State of Object

Questions 2.2.4 to 2.2.5

Create User Name

Display WiFi Codes

(4)

2.2.4 Write code for the **Create User Name** button that creates a user name by calling the **createUserName()** method. Write the user name to the user name textfield left blank in 2.2.3.

Paradise hotel WiFi

Question 2.2.2

Guest Details

Room Number: 103

Authenticate Guest

Question 2.2.3

Hotel Name: Paradise-Hotel

Surname: Maistry

Room Number: 103

Number of Nights: 1

User Name: Maistry103

Show State of Object

Questions 2.2.4 to 2.2.5

Create User Name

Display WiFi Codes

(3)

2.2.5 Write code for the **Display WiFi Codes** button that will:

- Obtain the number of devices that will be requiring WiFi access. Each room is allowed a maximum of 4 devices. Validate that the number entered in the text field is in the correct range. Continually prompt the user to enter the number of devices until a number in the correct range is entered.
- Create a password for each device by calling the appropriate method. Output the WiFi access details by calling the **toString()** method as indicated in the sample run.

(7)

TOTAL SECTION B: [55]

SECTION C**QUESTION 2: PROBLEM SOLVING****Scenario:**

Paradise Hotels keep records of the total booking for each resort for each season. The four resorts are Durban, Joburg, Cape Town and North West. The bookings for these resorts are separated into four holiday seasons.

The holiday seasons are as follows:

Low: Least popular period for holidays.

Medium: Medium demand Holidays (Certain public holidays and long weekends).

High: These are the most popular school holidays apart from peak season.

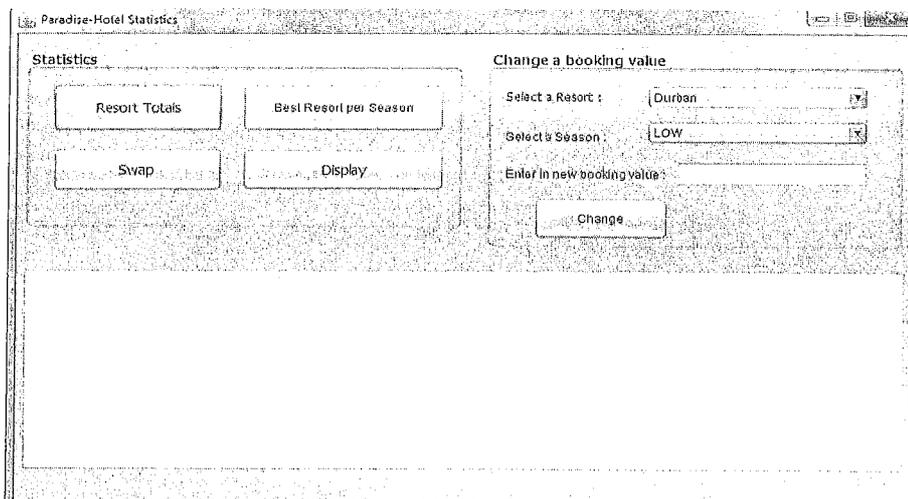
Peak: Most popular season – Christmas and New Year).

INSTRUCTIONS:

- The project Question3 is provided in the Data folder.
- Open the incomplete class called Question3.java in the folders Source Packages (src), Question3Package.

Do the following:

- Compile and execute the program. The program currently has no functionality. An example of the interface is given below.



3.1 Button: [Resort Totals]

Write code to do the following:

- Add the booking values for each resort (sum of each row) and store the total in a one-dimensional array called **resortTotals**. The array **resortTotals** has already been created for you.
- The following is an example of how this calculation is done:

	LOW	MED	HIGH	PEAK	Totals
Durban	150	250	295	350	1045
Joburg	120	220	250	279	869
Cape Town	160	278	315	410	1163
North West	100	168	201	245	714

The totals are obtained by adding the row of values for LOW, MED, HIGH and PEAK for each resort.

The total for Durban is calculated as follow:

$$(150+250+295+350 = 1045).$$

****NOTE:** There is no need to write code to display these totals in this button. All displays will be done in the Display button in 3.4. (6)

3.2 Button: [Best Resort per Season]

Write code to do the following:

- The best resort per season is calculated by finding out which resort had the highest booking for each season (highest value in each column). The following arrays have already been declared for you:
 - A one-dimensional array called **bestResortName** that must store the name of the resort that had the highest booking value for each season.
 - A one-dimensional array called **bestResortTotal** that must store the highest booking value for each season.
- The following is an example of how this calculation is done:

	LOW	MED	HIGH	PEAK
Durban	150	250	295	350
Joburg	120	220	250	279
Cape Town	160	278	315	410
North West	100	168	201	245
Best Resort	Cape Town	Cape Town	Cape Town	Cape Town
Best Total	160	278	315	410

The best resort is calculated by finding the highest value in each column. The highest value is stored in the array bestResortTotal and the name of the resort with the highest total is stored in the array bestResortName. The best resort for LOW season is CapeTown, which had the highest total of 160.

****NOTE:** There is no need to write code to display in this button. All displays will be done in the Display button in 3.4.

(11)

3.3 **Button: [Swap]**

After inspection of the bookings it was found that there has been an error in the values. The values for the seasons MED and HIGH have been mixed up and they have to be swapped.

Write code to do the following:

- The booking values in the MED column need to be swapped with the values in the HIGH column.
- The following is an example of the values before the swap button is pressed:

	LOW	MED	HIGH	PEAK
Durban	150	250	295	350
Joburg	120	220	250	279
Cape Town	160	278	315	410
North West	100	168	201	245

- The following is an example of the values **after** the swap button is pressed:

	LOW	MED	HIGH	PEAK
Durban	150	295	250	350
Joburg	120	250	220	279
Cape Town	160	315	278	410
North West	100	201	168	245

****NOTE:** There is no need to write code to display in this button. All displays will be done in the Display button in 3.4.

(5)

3.4 **Button: [Display]**

Write code to do the following:

- Display all values in the form of a grid in the text area provided.
- The display should resemble the following :

	LOW	MED	HIGH	PEAK	Totals
Durban	150	250	295	350	1045
Joburg	120	220	250	279	869
Cape Town	160	278	315	410	1163
North West	100	168	201	245	714
Best Resort	Cape Town	Cape Town	Cape Town	Cape Town	
Best Total	160	278	315	410	

****NOTE:** Marks will be awarded for displaying neatly in columns. The use of tabs or /and column formatting can be used.

(12)

3.5 **Button: [change]**

The change button allows you to change a booking value. You will be able To select a resort and a season from the combo boxes provided, and then Change the corresponding booking value with the new booking value entered in the text field.

Write code to do the following:

- Allow the user to select a resort and season form the combo boxes. Obtain the new booking value from the text field. Then change the corresponding value in the Two-dimensional array to this new booking value.

Example :

The following is Durban's booking values before the change :

	LOW	MED	HIGH	PEAK
Durban	150	250	295	350

Then the following entry is made and the button change is clicked :

Change a booking value

Select a Resort :

Select a Season :

Enter in new booking value :

The following is the booking values after the button change is clicked :

	LOW	MED	HIGH	PEAK
Durban	145	250	295	350

- Instantiate a new button (btnCalculate) alongside btnChange. btnCalculate must be 100 long and 45 wide as indicated in the sample run below.

Change a booking value

Select a Resort :

Select a Season :

Enter in new booking value :

- When btnCalculate is clicked, it must ask for the minimum rate(price) of a room in peak season of the resort that was selected and output the minimum amount of money that would have been generated from accommodation during peak season.

Change a booking value

Select a Resort :

Select a Season :

Enter in new booking value :

Change a booking value

Select a Resort :

Select a Season :

Enter in new booking value :

Input

Enter price per room?

Message

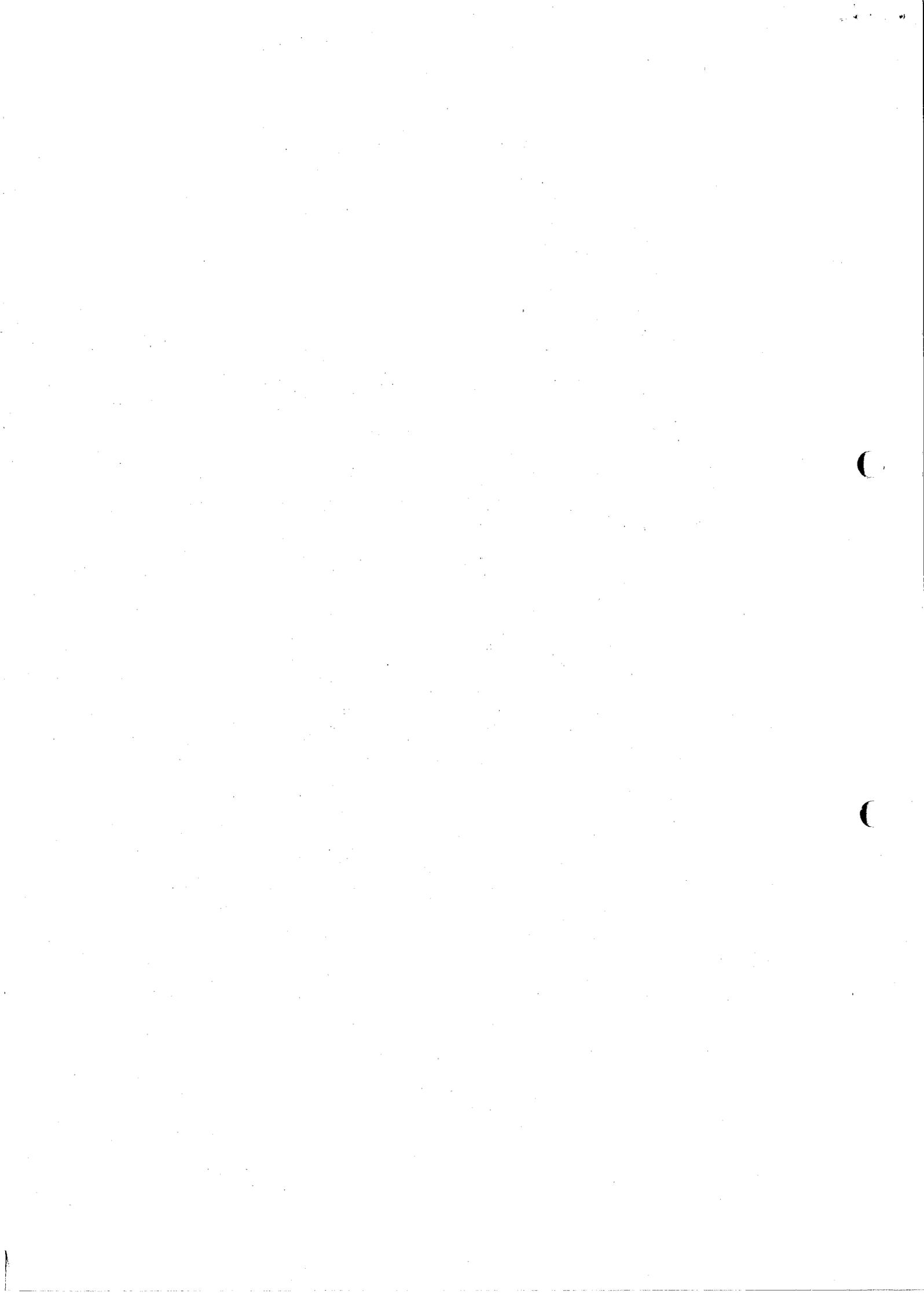
Total = 385000.0

(11)

- Enter your name as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required

TOTAL SECTION C: [45]

GRAND TOTAL: 150



IT P1 – KZN COMMON PAPER JUNE 2016 : MEMO

Question 1

package Question1Package;

DecimalFormat df = new DecimalFormat("R 0.00");

String arrCountryNames[] = {"USA", "EUROPE", "BRITAIN", "INDIA", "AUSTRALIA"};

double arrConversions[] = {0.066799, 0.058111, 0.046081, 4.451249, 0.089593};

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

//Question 1.1

String sName = txtID.getText(); ✓

String nam = txtName.getText(); ✓

txtAWelcome.append("Welcome to Hilltop Hotel "+nam+" "+sName+""); ✓

txtBWelcome.append("\n ✓ We hope your stay with us is memorable");

txtCWelcome.append("\nPlease contact reception on 031-2089990");

txtDWelcome.append("\nfor any assistance");

} (5)

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

// Question 1.2

String sName = txtSurname.getText();

String nam = txtName.getText();

String ID = txtID.getText(); ✓

int year = (1900+ Integer.parseInt (ID.substring(0,2)));

int age = 2016 - year; ✓

String guestNumber =

sName.substring(0,1).toUpperCase()+"sName.substring(1,3) "+age ✓

+nam.substring(nam.length()-3,nam.length()); ✓

txtGuestID.setText(guestNumber); ✓

} (10)

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

//Question 1.3

String accomType = (String)(cmbAccomType.getSelectedItem()); ✓

int numNights = Integer.parseInt(txtNumNights.getText()); ✓

int numGuests = Integer.parseInt(txtNumGuests.getText());

double cost = 0;

switch (accomType.charAt(0)) {

case 'S':{cost = 1000; break;} ✓

case 'F':{cost = 1500; break;} ✓

case 'E':{cost = 2500; break;} ✓

} cost = cost*numNights; ✓

cost = cost*numNights; ✓

```
✓ if (chbBreakfast.isSelected()) ✓
{
    cost. += (250*numNights*numGuests);
}
boolean flag=true; ✓
if (rtnCash.isSelected()) ✓
{
    txtPayment.setText(df.format( cost));
}
else
{
    String cardNum = JOptionPane.showInputDialog("Enter card number"); ✓
    if (cardNum.length()==10) ✓
    {
        flag = true;
        for(int i =0;i<10;i++) ✓
        {
            if (!Character.isDigit(cardNum.charAt(i))) ✓
            {
                flag = false; ✓
            }
        }
    }
    else
    {
        flag = false; ✓
    }
    if (flag) ✓
    {
        cost = cost*1.05; ✓
        txtPayment.setText(df.format(cost));
    }
    else
    {
        JOptionPane.showMessageDialog(null, "Invalid card number"); ✓
        rtnCash.setSelected(true); ✓
        txtPayment.setText(df.format(cost)); ✓
    }
}
}
```

```

private void btnCalculateActionPerformed(java.awt.event.ActionEvent evt) {
//Question 1.4
double rand = 0;
String country = (String)(cmbCountry.getSelectedItem());
for (int i = 0; i<arrConversions.length; i++)
{
if (country.equals(arrCountryNames[i]))
{
rands = (Double.parseDouble(btnAmount.getText())) * arrConversions[i];
}
}
btnRandValue.setText(df.format(rands));
}

```

(10)

```

Question 2
package QuestionTwo;
/**
 *OOP June 2016
 */
public class WiFiGenerator {

```

```

//2.1.1
private String hotelName;
private String surname;
private int roomNumber;
private int numDays;
private String userName;
private String password;

```

(4)

```

//2.1.2
public WiFiGenerator(String hotelName, String surname, int roomNumber, int
numDays)
{

```

```

this.hotelName = hotelName;
this.surname = surname;
this.roomNumber = roomNumber;
this.numDays = numDays;
userName = "";
password = "";
}

```

(4)

```

//2.1.3
public String getHotelName() {
return hotelName;
}

```

```

public String getSurname() {
return surname;
}

```

```

public int getRoomNumber() {
return roomNumber;
}

```

```

public int getNumDays() {
return numDays;
}

```

(for all getters)

```

public String getUsername() {
    return userName;
}

public String getPassword() {
    return password;
}

public void setUsername(String userName) {
    this.userName = userName;
}

public void setRoomNumber(int roomNumber) {
    this.roomNumber = roomNumber;
}

//2.1.4
public void createUserUsername() (uncommenting)
{
    userName = surname+roomNumber;
}

//2.1.5
public void createPassword()
{
    int count = 0;
    password = "";
    while (count<4)
    {
        int pos = (int)(Math.random()*hotelName.length());
        password += hotelName.charAt(pos);
        count++;
    }
    password += numDays*24;
}

//2.1.6
public String toString()
{
    String report = hotelName+" Free WiFi Login\n";
    report += "Room number: "+roomNumber+"\n";
    report += "User Name: "+userName+"\n";
    report += "WiFi Password: "+password;
    return report;
}
}

```

```

//2.2.1
WiFiGenerator objWifi;

private void btnCreateObjectActionPerformed(java.awt.event.ActionEvent evt) {
//2.2.2
    boolean flag = true;
    int room = (int)(cbxRoomNo.getSelectedItem());
    File tFile = new File ("wifiLog.txt");
    if (tFile.exists())
    {
        JOptionPane.showMessageDialog(null, "File does not exist!");
        System.exit(0);
    }
    try
    {
        Scanner fr = new Scanner(tFile);
        String hotelName = fr.nextLine().substring(0, 14);
        while (fr.hasNext())
        {
            String line = fr.nextLine();
            String [] part = line.split("#");
            int roomNo = Integer.parseInt(part[1]);
            int nights = Integer.parseInt(part[2]);
            if (roomNo==room)
            {
                flag = false;
                objWifi = new WiFiGenerator(hotelName, part[0], roomNo, nights);
            }
        }
        if (flag)
        {
            JOptionPane.showMessageDialog(null, "room not booked");
        }
        else
        {
            btnShowState.setEnabled(true);
        }
    }
    fr.close();
}
catch (FileNotFoundException ex)
{
    Logger.getLogger(WIFIGUI.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

```

private void btnStateOfObjActionPerformed(java.awt.event.ActionEvent evt) {
// 2.2.3
    txtHotelName.setText(objWifi.getHotelName()); ✓
    txtRoomNumber.setText(objWifi.getRoomNumber()+""); ✓
    txtNumNights.setText(objWifi.getNumDays()+""); ✓
    txtSurname.setText(objWifi.getSurname()); ✓
}
(4)

private void btnUserNameActionPerformed(java.awt.event.ActionEvent evt) {
// 2.2.4
    objWifi.createUser(); ✓
    txtUserName.setText(objWifi.getUserName()); ✓
}
(3)

// 2.2.5
private void btnWiFiCodesActionPerformed(java.awt.event.ActionEvent evt) {
    int numDevices=Integer.parseInt(JOptionPane.showInputDialog(null,"Enter
    number of devices ")); ✓
    while( numDevices<1 || numDevices>4) ✓
    {
        numDevices=Integer.parseInt(JOptionPane.showInputDialog(null,"Enter number
        of devices ")); ✓
    }
    txtOutput.setText("");
    for (int i = 0; i < numDevices; i++) ✓
    {
        objWifi.createPassword(); ✓
        txtOutput.append(objWifi.toString()+"\n\n");
    }
}
(7)

```

Question 3
package Question3Package;

```

public class Question3 extends javax.swing.JFrame {
    int [][][] bookings = {{150,250,295,350},
                          {120,220,250,279},
                          {160,278,315,410},
                          {100,168,201,245}};

    String [] resorts = {"Durban","Joburg","Cape Town","North West"};
    String [] seasons = {"LOW","MED","HIGH","PEAK"};
    int [] resortTotals = new int [4];
    String [] bestResortName = new String[4];
    int [] bestResortTotal = new int[4];

    public Question3() {
        initComponents();
        setLocationRelativeTo(null);
        for (int i = 0; i < 4; i++)
        {
            cmbResorts.addItem(resorts[i]);
            cmbSeasons.addItem(seasons[i]);
        }
    }
//=====
// Question 3.1
//=====
private void btnResortTotalsActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    int sum;
    for (int r = 0; r < 4; r++) ✓
    {
        sum=0; ✓
        for (int c = 0; c < 4; c++) ✓
        {
            sum+=bookings[r][c]; ✓
        }
        resortTotals[r]=sum; ✓
        alternately: resortTotals[r] +=bookings[r][c]; ✓ ✓
    }
}
(6)

```



```
//=====
//Question 3.5
//=====
Private void btnChangeActionPerformed(java.awt.event.ActionEvent evt) {

    int row = cbxResorts.getSelectedIndex(); ✓
    int col = cbxSeasons.getSelectedIndex(); ✓
    int newValue = Integer.parseInt( txtNewBookingValue.getText()); ✓
    bookings[row][col] =newValue; ✓

    JButton btnCalculate = new JButton(); ✓
    btnCalculate.setEnabled(true); ✓
    btnCalculate.setText("Calculate"); ✓
    pnlChangeBooking.add(btnCalculate); ✓
    pnlChangeBooking.show(true); // show the button
    btnCalculate.setBounds(btnChange.getX()+150, btnChange.getY(), 120, 45); ✓

    btnCalculate.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent ae) {
            double price = Double.parseDouble(OptionPane.showInputDialog("Enter
            price per room?")); ✓
            JOptionPane.showMessageDialog(null, "Total = "+price*bookings[row][3] ✓);
        }
    });
}

(11)
```

ANNEXURE A
LEARNER NAME: _____

QUESTION 1: MARKING GRID – GENERAL PROGRAMMING SKILLS

NAME OF LEARNER	GRADE MAX MARKS	LEARNER MARK
1.1 BUTTON – [Question1.1] Extract surname ✓ Extract name ✓ Display welcome message in text area on different lines ✓✓✓	5	
1.2 BUTTON – [Question1.2] Extract first letter and convert to uppercase ✓ Extract next 2 letters of surname ✓ Calculate age ✓✓✓ Extract the last 3 characters of the name ✓✓ Concatenate the characters and display in textfield ✓✓✓	10	
1.3 BUTTON – [Question1.3] Extract the number of people from textbox as integer ✓ Extract number of nights from textbox as integer ✓ Extract type of accommodation from combo box ✓ Case/Switch/If (accommodation type) and allocate correct cost to cost variable ✓✓ Multiply cost by number of nights ✓ Check if breakfast has been selected ✓ Multiply 250 by no. of people and number of nights and add to cost ✓✓ Check if radio button for Card is selected ✓ Use a dialog box to enter a card number. ✓ Set flag to true ✓ if length of card number is not 10 – set flag to false ✓✓ Check if all characters are digits: Loop through the characters of the card ✓✓ Use method to check if each character is a digit ✓✓ if not a digit – set flag to false ✓ if card number is valid add 5% to the total cost ✓✓ if card number is not valid Use a display dialog box to display "Invalid card number" ✓ Set radio button selection to cash option ✓ Show formatted total cost in text box ✓✓	25	

1.4	BUTTON – [Question1.4] Extract country from combo box ✓ Loop through the country array ✓✓ if the country from the combo box = the country in the array ✓ calculate the conversion ✓✓✓ Display the formatted conversion in the textbox ✓	10
-----	--	----

QUESTION 2: MARKING GRID - OOP

		MAX MARKS	LEARNER MARK
2.1.1	private ✓ hotelName, surname - String ✓ roomNumber, numDays- int ✓ username, password - String ✓ Correct parameters ✓	4	
2.1.2	assigning values correctly: hotelName, surname ✓ roomNumber, numDays ✓ username, password ✓	4	
2.1.3	6 Assessor methods ✓✓ correct 2 mutator methods ✓✓	4	
2.1.4	Removal of comments ✓	1	
2.1.5	set count to 0 ✓ set password to an empty string ✓ loop 4 times (use of while loop only) ✓ set pos to a random number from 0 to number of letters in the hotel name. ✓ attach letter at position pos in the hotel name to password. ✓ increase count by 1 ✓ multiply the number of nights by 24 ✓ set the password attribute in the WiFiGenerator.java class to the generated password ✓	8	
2.1.6	1 mark for each Correct output statement and attribute ✓✓✓✓ Correct layout (use of \n and \t) ✓	5	
2.2.1	WiFiGenerator objWifi; ✓	1	

2.2.2	<p>Button: [Authenticate Guest]</p> <p>Obtain room no. and convert to integer ✓ Create file object and check if file exists ✓ suitable message terminate program ✓ if don't exist in correct part of if..statement Create scanner object ✓ Read first line and obtain hotel name ✓ While loop with hasNext() ✓ read subsequent lines ✓ and break up into fields using delimiter ✓ (Split or Scanner) Change to correct data types ✓ use of boolean value to flag if room number is in file and output of not found message ✓ Check if room number matches room number search ✓ instantiate object if room number is found ✓</p>	14	
2.2.3	<p>Enable btnShowState ✓ close file ✓</p> <p>Button: [Show State of Object]</p> <p>Hotel name ✓, surname ✓, room number ✓, number of nights ✓ correctly called using accessor methods</p>	4	
2.2.4	<p>Button: [Create User Name]</p> <p>Call createUserName using object ✓ setText ✓ call username method using object ✓</p>	3	
2.2.5	<p>Button: [Display WiFi Codes]</p> <p>Set number of devices to 0 ✓ Get number of devices and convert to int ✓ range check upper ✓ lower ✓ run loop ✓ call CreatePassword method ✓ call toString method ✓</p>	7	
		[55]	

QUESTION THREE – OPEN ENDED

		Max Mark	Learner Mark
3.1.	<p>Button : [Resort Totals]</p> <p>Loop through row 0 to 3 ✓ Set sum to 0 ✓ Loop through col 0 to 3 ✓ Add booking value from 2D Array at position [row][col] to sum ✓ ✓ End col loop Set the value in the 1D Array resortTotal at position [j] to sum ✓ End row loop</p>	6	
3.2.	<p>Button : [Best Resort]</p> <p>Loop through row 0 to 3 ✓ Set bestName to "" ✓ Set bestTotal to 0 ✓ Loop through col 0 to 3 ✓ If ✓ booking value at pos [col][row] > ✓ bestTotal ✓ Set BestTotal to the booking value from the 2D Array at position [col][row] ✓ Set BestName to the Resort Name from the 1D Array at position [row] ✓ End col loop Set the value in the 1D Array bestResortName at position [j] to bestName ✓ Set the value in the 1D Array bestTotal at position [j] to bestTotal ✓ End row loop</p>	11	
3.3.	<p>Button : [Swap]</p> <p>Loop through index 0 to 3 ✓ Set temp to the booking value from the 2D array at position [index] ✓ (this mark is for keeping the row index constant for all 3 steps) [1] ✓ Set the booking value from 2D array at position [index][1] to the booking value from 2D array at position [index][2] ✓ Set the booking value from 2D array at position [index][2] to temp. ✓ End row loop</p>	5	

3.4.	<p>Button : [Display] txaOutput.setText ✓ display the season from the 1D array seasons [0].....[3] on the first line. ✓ Loop through row 0 to 3 ✓ txaOutput.append ✓ display the resort name from the 1D array resorts at position [row] ✓ Loop through col 0 to 3 ✓ txaOutput.append display the booking value in the 2D array booking value at pos [col][row] ✓ ✓ End col loop txaOutput.append display the resort total from the 1D array Resort Totals at position [row] ✓ End row loop txaOutput.append display the names from the 1D array bestResortNames [0].....[3] on the same line. ✓ txaOutput.append display the value from the 1D array bestResortTotal [0].....[3] on the same line. ✓ For Displaying in proper Grid pattern as shown in the example, using either tabs or string formatting. ✓</p>	12	
3.5.	<p>Button : [Change] Set row and col to the selected index value from the resort combo box and seasons combo box ✓ Set newValue to the value obtained from the textbox and Convert newValue to integer ✓ Set the booking value from 2D array at position [row][col] ✓ to newValue. ✓ instantiate a new button ✓ enable button ✓ add caption ✓ Add button to panel changeBooking ✓ Correct boundaries for button and size of button ✓ Action listener – room price ✓ Output price ✓</p>	11	[45]

