MEMORANDUM

Programme	**Diploma In Information Technology**
Module	**Programming 2B**
Module Code	**PRG220**
Module NQF Level	**6**
Credits	**15**
Test/Exam	**Supplementary Exam Memo**
Semester	**2nd**
Date Written	

Total marks	**100**
Duration	**2 Hours**
Pass mark	**50%**
Weighting	**50%**
Examiner	**Frans Rampai**
Moderator	**Zukile Ndyalivana**

This question paper consists of **15** pages including the cover page.

REQUIREMENTS:

Learner Requirements: Stationery and Examination Answer booklet
Equipment Requirements: None

This paper consists of:

1.	Section A:	**50 marks**
2.	Section B:	**30 marks**
3.	Section C:	**20 marks**

ALL sections are **COMPULSORY**. It is in your own interest to write legibly and to present your work neatly.

PLEASE READ THE ASSESSMENT RULES AND REGULATIONS THAT FOLLOW

Learners are warned that contravening any of the examination rules or disobeying the instructions of an invigilator could result in the examination being declared invalid. Disciplinary measures will be taken which may result in the students' expulsion from Damelin.

ASSESSMENT RULES AND REGULATIONS

Please ensure that you have read and fully understand the following assessment rules and regulations prior to commencing with your assessment:

1. To be permitted access to the examination, a learner must arrive with:
   - an Identity Document or other official proof of identity (for example,
   - a student card, passport or driver's licence card with photo); and
   - the required exam stationery.
2. No learner may enter the examination room more than 30 minutes after the examination sitting has commenced and no candidate may leave the room less than one hour after the examination sitting has commenced.
3. No extra time will be allowed should a student arrive late.
4. All learners must sign the *Attendance Register* for the examination on arrival.
5. It is the responsibility of learners to familiarise themselves with the examination rules prior to sitting for the examination.
6. All examinations are to be written on the date and time officially stipulated by the College.
7. It is the responsibility of learners to ensure that they are writing the correct paper and that the question paper is complete
8. Cell phones must be switched *off* prior to entering the exam venue. Cell phones and wallets may be placed under candidates' chairs rather than at the front of the room.
9. Learners may not handle cell phones or wallets during the exam.
10. No weapon of any description may be taken into the assessment room.
11. All personal belongings are to be placed at the front of the examination room. Personal belongings brought to the examination are at the owner's risk.
12. Smoking is not permitted and learners will not be allowed to leave the examination room in order to smoke
13. Once the examination has commenced, all conversation of any form between candidates must cease until after candidates have left the room, after the examination.
14. *Only* the official College examination book, as supplied by the College, may be used.
15. Learners must ensure that their student number is written on the answer book.
16. Learners are responsible for ensuring that they follow the instructions in the examination for submitting their answers.
17. Please read the instruction appearing on the examination paper carefully
18. The number of every question must be clearly indicated at the top of every answer.
19. No pages may be torn out of the answer book. All question papers and scrap paper must be handed to the invigilator after the examination.
20. Learners finishing earlier are to leave the examination room as quietly as possible on the instruction of the invigilator and may not talk until outside the building where the examination is being written.
21. Only under exceptional circumstances will a learner be permitted to leave the examination room during the examination, and if the invigilator gives permission. An invigilator must accompany the learner. Only one learner at a time may be absent from the examination room.
22. Candidates may not act dishonestly in any respect.

SECTION A:
MULTIPLE CHOICE QUESTIONS
Select the correct answer from the options available.
In the provided answer book, write down the question number and the letter of the correct answer next to it.
For example: 1.1 D
Each question is worth one (1) marks.

QUESTION 1 **(10 Marks)**

Outcomes:
- Sort array elements using the bubble sort algorithm
- Sort array elements using the insertion sort algorithm
- Use two-dimensional and other multidimensional arrays
- Use the Arrays class
- Use the ArrayList class

1.1	When you place objects in order beginning with the object with the highest value, you are sorting in _____ order.		(1)
	A.	Acquieiscing	
	B.	Ascending	
	C.	Demeaning	
	D.	Descending	
ANS: D√			
1.2	Using a bubble sort involves _____.		(1)
	A.	Comparing parallel arrays	
	B.	Comparing each array element to the average	
	C.	Comparing each array element to the adjacent array element	
	D.	Performing the comparison-swap procedure of adjacent array elements.	
ANS: D√			
1.3	In the following array, what is the value of address [1][1]? String address = { {“123 Oak”,”345 Elm”}, {“87 Maple”,”901 Linden”} };		(1)

	A.	"123 Oak"	
	B.	"345 Elm"	
	C.	"87 Maple"	
	D.	"901 Linden"	

ANS: D√

1.4	A_____array has rows of different lengths.		(1)
	A.	Ragged	
	B.	Two-dimensional	
	C.	Three dimensional	
	D.	One-dimensional	

ANS:A√

1.5	When array elements are objects, you usually want to sort based on a particular _____ of the object.		(1)
	A.	Field	
	B.	Method	
	C.	Name	
	D.	Type	

ANS:A√

1.6	The following defines a _____array: Int [][] nums = {{1,2},{2,4},{5,6}};		(1)
	A.	One-dimensional	
	B.	Two-dimensional	
	C.	Three-dimensional	
	D.	Six-dimensional	

ANS:B√

1.7	If the value of credits [0].length is not equal to credits [1].length, you know credits is a_____ array.		(1)
	A.	three-dimensional	
	B.	One-dimensional	
	C.	Two-dimensional	
	D.	jagged	

ANS:D√

1.8	Which of the following is a requirement when you use a binary search method with an array?		(1)
	A.	The array must be numeric	
	B.	The array must have been sorted in ascending order.	
	C.	The array must have at least three elements.	
	D.	None of the above	

ANS:B√

1.9	The chief advantage to using the ArrayList class instead of the Arrays class is that an ArrayList _____.		(1)
	A.	Can be much convenient to use	

	B.	Is easier to search	
	C.	Is dynamically resizable	
	D.	Can be used as an argument to a static method.	
ANS:C√			
1.10	An advantage to suing an enumerated data type is _____.		(1)
	A.	Errors are reduced because only a limited set of values can be used with the type.	
	B.	Time is saved because programs with enumerated types compile faster	
	C.	Coding time is reduced because enumerated types are created automatically by the compiler.	
	D.	All of the above are true.	
ANS:A√			

TRUE / FALSE QUESTIONS

Select the correct answer from the options available. Indicate your answer on the Answer book.

provided. For example: 2.1 FALSE

Each question is worth two (2) marks.

QUESTION 2 (10 Marks)

Outcomes:

- Learn about the concept of inheritance
- Extend classes

2.1	Employing inheritance reduces errors because many of the methods you need have already been used and tested.	(2)
	ANS: True√√	
2.2	Derived class is also known as a parent class.	(2)
	ANS:False√√	
2.3	Extends is the only keyword to create inheritance in java.	(2)
	ANS: True√√	
2.4	Polymorphism is the concept of keeping data private in java.	(2)
	ANS: False√√	
2.5	If you declare a data field or method that is protected, it can be used within its own class or in any class inherited from that class.	(2)
	ANS: True√√	

MATCHING

Match the items in column A to corresponding descriptions in column B.
In the provided answer book, write down the question number and the letter of the correct answer next to it.
For example: 3.1 D
Each question is worth one (1) mark.

QUESTION 3 **(10 Marks)**

Outcomes:

- Create and use abstract classes
- Use dynamic method binding
- Create arrays of subclass objects
- Use the Object class and its methods
- Use inheritance to achieve good software design
- Create and use interfaces
- Create and use packages

	Column A		**Column B**
3.1	abstract class	**A.**	An application's ability to select the correct subclass method
3.2	Subclass of abstract class	**B.**	Variable declared within parenthesis of a method
3.3	Dynamic method binding	**C.**	Inherits the abstract method from its parent
3.4	parameter	**D.**	Wakes up a single thread that is waiting on this object's monitor.
3.5	void notify()	**E.**	Prohibited in java
3.6	toString()	**F.**	A named collection of classes.
3.7	Equals()	**G.**	Cannot create any concrete objects
3.8	Multiple inheritance	**H.**	Contains information about the object
3.9	interface	**I.**	Returns a Boolean value
3.10	package	**J**	Description of what a class does

ANS:

3.1 G √ 32.C√ 3.3 A√ 3.4 B√ 3.5 D√ 3.6 H√ 3.7 I√ 3.8 E√ 3.9 J√ 3.10F√

QUESTION 4 **(20 Marks)**

Outcomes:

- Use two-dimensional and other multidimensional arrays
- Use the Arrays class

• Use the ArrayList class

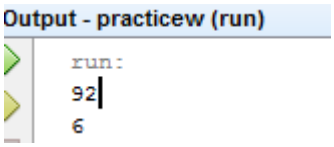
Analyze the code snippets below and write the output:

4.1 (5)

```
package practicew;
public class Practicew {
    public static void main(String[] args) {
        int [] arr={34,45,6,7};
        System.out.println(sum(arr));
        System.out.println(min(arr));
    }
static int sum(int[] a) {
   int total = 0;
   for (int i : a)
      total += i;
   return total;
}
static int min(int[] a) {
   int least = a[0];
   for (int i : a)
      if (i < least)
         least = i;
   return least;
}
}
```

ANS:

√√√√√



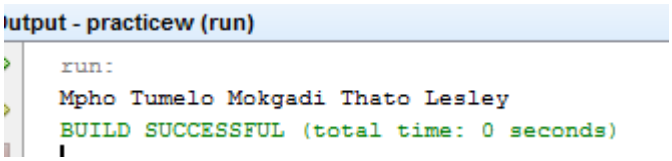
4.2 (5)

```
package practicew;
public class Practicew {
    public static void main(String[] args) {
        String [] arr={"Mpho","Tumelo","Mokgadi","Thato","Lesley"};
        int [] arr1={45,56,23,12};
        System.out.println(joinUp(arr));
    }
static String joinUp(String[] a) {
```

```
    String s = "";
    for (int i = 0; i < a.length; i++) {
      s = s + a[i];
      if (i < a.length-1)
        s = s + " ";
    }
    return s;
}
}
```

ANS:

√√√√√

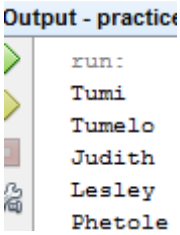


4.3 (5)

```
package practicew;
public class Practicew {
   public static void main(String[] args) {
      String [][] arr={new String[]{"Mpho","Tumelo",
                  "Mokgadi","Thato","Lesley"},new String[]{"Tumi",
                  "Phetole","Jan"},new String[]{"Judith",
                  "Nelly"} };
       System.out.println(arr[1][0]);
       System.out.println(arr[0][1]);
       System.out.println(arr[2][0]);
       System.out.println(arr[0][4]);
       System.out.println(arr[1][1]);
   }
}
```

ANS:

√√√√√

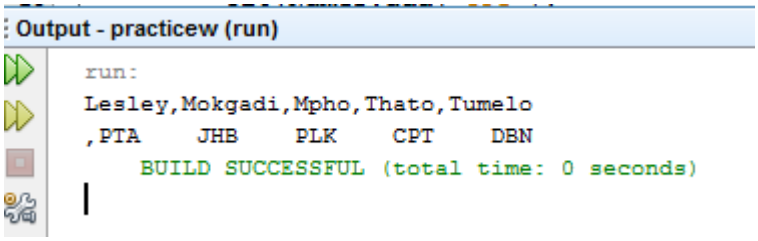


4.4 package practicew; (5)

```
import java.util.ArrayList;
import java.util.Arrays;
public class Practicew {
    public static void main(String[] args) {
        String [] arr={"Mpho","Tumelo\n",
                    "Mokgadi","Thato","Lesley"};
        ArrayList CityNames=new ArrayList();
        CityNames.add("PTA");
        CityNames.add("JHB");
        CityNames.add("PLK");
        CityNames.add("CPT");
        CityNames.add("DBN\n");
        Arrays.sort(arr);
        for(int i=0;i<arr.length;i++)
        System.out.print(arr[i]+",");
         for(int i=0;i<CityNames.size();i++)
         System.out.print(CityNames.get(i) + "    ");
    }
}
```

ANS:

√√√√√



SECTION B:

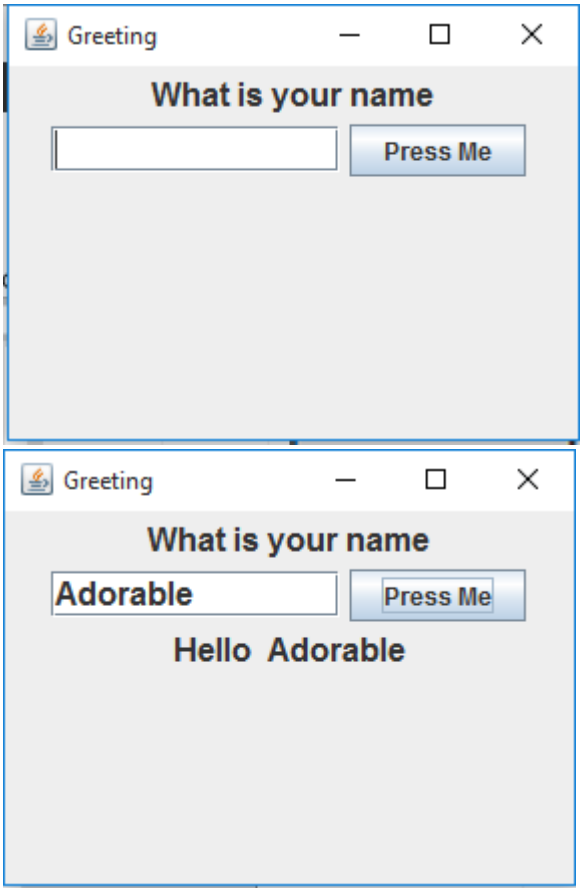
QUESTION 5 **(30 Marks)**

Outcomes:

- Understand Swing components
- Use the JFrame class
- Use the JLabel class
- Use a layout manager
- Extend the JFrame class
- Add JTextFields, JButtons, and tool tips to a JFrame
- Learn about event-driven programming
- Understand Swing event listeners

• Use the JCheckBox, ButtonGroup, and JComboBox classes

• Use content panes
• Use colour
• Learn more about layout managers
• Use JPanels to increase layout options
• Create JScrollPanes
• Understand events and event handling more thoroughly
• Use the AWTEvent class methods
• Handle mouse events
• Use menus

5.1	With the use of JFrame create an application that contains a text box and a button such that when you press button “Press me” then The text entered will be displayed on a label on the form as is on below interfaces. **Note: marks are only awarded to code thus do not drag and drop.** 	(15)
	To markers: **since code may differ please allocate marks using won discretion.** ANS: /* * To change this license header, choose License Headers in Project Properties. * To change this template file, choose Tools | Templates * and open the template in the editor. */ package question5point1;	

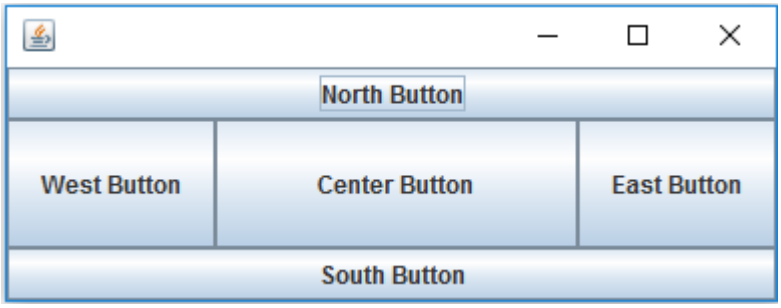
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Question5point1 extends JFrame implements ActionListener {
    /**
     * @param args the command line arguments
     */
    JLabel lbl_title=new JLabel("What is your name");
Font bigFont=new Font("Arial",Font.BOLD,16);
JTextField txt_txtb=new JTextField(10);
JButton btn_pressMe=new JButton("Press Me");
JLabel lbl_result=new JLabel("");
final int width=300;
final int height=225;

public Question5point1()
{
    super("Greeting");
    setSize(width,height);
    setLayout(new FlowLayout());
    lbl_result.setFont(bigFont);
    txt_txtb.setFont(bigFont);
    lbl_title.setFont(bigFont);


add(lbl_title);
add(txt_txtb);
add(btn_pressMe);
add(lbl_result);

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
btn_pressMe.addActionListener(this);
}
    public static void main(String[] args) {
   Question5point1 frame=new Question5point1();
   frame.setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        lbl_result.setText("Hello  "  +txt_txtb.getText());
    }
}
```

5.2	**Using BorderLayout** The BorderLayout manager is the default manager class for all content panes. You can use the BorderLayout class with any container that has five or fewer components. (However, any of the components could be a container that holds even more components.) When you use the BorderLayout	(15)

manager, the components fill the screen in five regions: north, south, east, west, and center. A JFrame that contains five JButton objects that fill the five regions in a content pane that uses BorderLayout.

Note: marks are only awarded to code thus do not drag and drop.

With use of BoarderLayout write an application that produce an interface below.



ANS:
To markers:
since code may differ please allocate marks using won discretion.

```
package question5;
import javax.swing.*;
import java.awt.*;
public class Question5 extends JFrame {
    private final JButton nb=new JButton("North Button");
     private final JButton sb=new JButton("South Button");
      private final JButton eb=new JButton("East Button");
      private final JButton wb=new JButton("West Button");
       private final JButton cb=new JButton("Center Button");
      private final Container con=getContentPane ();

      public Question5()
      {
         con.setLayout(new BorderLayout());
         con.add(nb,BorderLayout.NORTH);
         con.add(sb,BorderLayout.SOUTH);
         con.add(eb,BorderLayout.EAST);
         con.add(wb,BorderLayout.WEST);
         con.add(cb,BorderLayout.CENTER);
         setSize(400,150);
      }
```

```
    public static void main(String[] args) {
      Question5 frame=new Question5();
      frame.setVisible(true);

    }

}
```

SECTION C:

QUESTION 6 **(20 Marks)**

Outcomes:

- Learn about exceptions
- Try code and catch exceptions
- Throw and catch multiple exceptions

Short Case Study

An exception is an unexpected or error condition. The programs you write can generate many types of potential exceptions, such as when you do the following:

- You issue a command to read a file from a disk, but the file does not exist there.
- You attempt to write data to a disk, but the disk is full or unformatted.
- Your program asks for user input, but the user enters invalid data.
- The program attempts to divide a value by 0.
- The program tries to access an array with a subscript that is too large or too small.

These errors are called exceptions because, presumably, they are not usual occurrences; they are "exceptional." Exception handling is the name for the object-oriented techniques to manage such errors. Unplanned exceptions that occur during a program's execution are also called runtime exceptions. Java has two basic classes of errors: Error and Exception. Both classes descend from the Throwable class. Like instantiations of all other classes in Java, exceptions originally descend from Object.

6.1 Write an application named BadSubscriptCaught in which you declare an array of 10 test marks as follows:

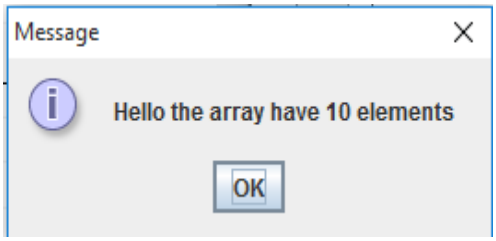
int marks[] = { 40, 50, 60 ,50,28,52,67,89,98,67 }; .

Write a try block in which you prompt the user of an integer and display the Test mark in the requested position. Create a catch block that catches the potential ArrayIndexOutBoundsException thrown when the user enters a

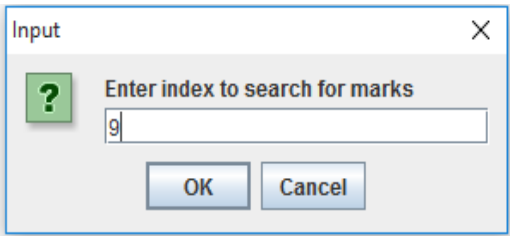
number that is out of range. The catch block should also display an error message. Name the file as **BadSubscriptCaught.java.**

Expected output:

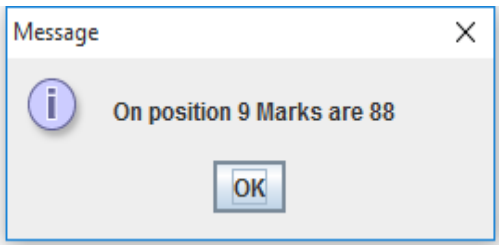
1.



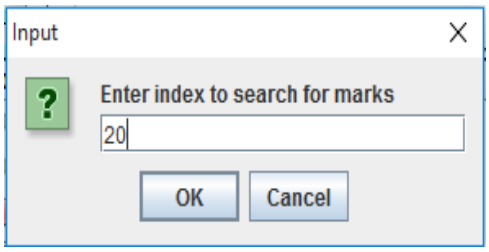
2.



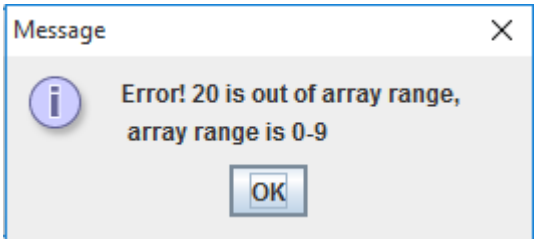
3.



4.



5.



To markers:

Since code may differ please use own discretion when allocating ticks.

ANS:

```
package badsubscriptcaught;
import javax.swing.JOptionPane;
public class BadSubscriptCaught {
    public static void main(String[] args) {
        int index=0;
            int marks[] = { 77, 70, 60 ,50,28,52,66,89,92,88 };

    JOptionPane.showMessageDialog(null,"Hello the array have 10 elements ");
```

```
    try
    {
      index=Integer.parseInt(JOptionPane.showInputDialog("Enter index to
search for marks"));
      int m1 = marks[index];
       JOptionPane.showMessageDialog(null,"On position "+index+ " Marks are "
+ m1);
    }
    catch(ArrayIndexOutOfBoundsException e)
    {
      JOptionPane.showMessageDialog(null,"Error! "+index +" is out of array
range,"
              + "\n array range is 0-9 ");
    }
  }
}
```

∞*End of Question Paper*∞