

Please fill in your Student Number and Name.

Student Number : _____

Name: _____

Student Number: _____

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2007

Supplementary Examination

Question	Max	Internal	External
1	25		
2	0		
3	0		
4	0		
5	5		
6	5		
7	10		
8	0		
9	5		
10	15		
TOTAL	100		

Marks : 100

Time : 3 hours

Instructions:

- a) Answer all questions.
- b) Write your answers in the space provided.
- c) Show all calculations where applicable.

Section A: Recursion, Files and Exceptions

Question 1: Recursion, Files and Exceptions [25 marks]

Examine the following Java application and answer the questions that follow.

```
import java.io.*;
public class Exam2Demol
{
    public static void main(String[] args) throws
        FileNotFoundException, KeyNotFoundException
    {
        int [] arr = {1,5,8,57,100,119,300,401,1000,12000};
        PrintWriter printW = new PrintWriter(new
            FileOutputStream("fileA.txt"));
        int p = look(arr,0,10,100);
        printW.println(p);
    }

    public static int look(int[] a, int first, int last, int key)
        throws KeyNotFoundException
    {
        int mid,result=0;
        boolean found = false;
        while ( (first <= last) && !(found) )
        {
            mid = (first + last)/2;
            if (key == a[mid])
            {
                found = true;
                result = mid;
            }
            else if (key < a[mid]) last = mid - 1;
            else if (key > a[mid]) first = mid + 1;
        }
        if (first > last) throw new KeyNotFoundException(key);
        return result;
    }
}
```

a) Describe in clear English what this program does.

[2]

b) The program above does not close any of the streams that it opens. Explain clearly why this can be a problem. [3]

c) Give an example of the use of an **anonymous object** in the program above. [1]

d) What is a checked exception in Java? [1]

e) Explain why it would **not** have been sensible for the Java designers to make a `StackOverflowException` a **checked** exception. [2]

f) Rewrite the main method so that all the checked exceptions it generates are caught and handled with suitable messages. [5]

g) Write a suitable definition for the `KeyNotPresentException` class, including **all** necessary constructors. [6]

h) The application could also be written to use a recursive definition of the `look` method. Give one advantage and one disadvantage of doing this. [2]

i) Now write a **recursive** method

`public static void reverse(int[] a, int first, int last)`
for the class `Exam2Demo`. This method will print to the screen the array `a[]` in reverse order. For example, for the array `a[]` given above, the output will be:

16 14 12 10 8 6 4 2 0 -2

Note that **no marks** will be given for iterative solutions. [3]

```
public static void reverse(int[] a, int first, int last)
{
```

```
}
```

Section B: UML, Interfaces and Generics

Question 2: UML [9 marks]

Draw a UML class diagram to describe the relationship between classes in an information management system of a medical clinic. In the system, there are 2 main kinds of users. The first one is health care professionals (e.g., doctors and nurses) and the second one is patients. Each user signs up with a medical scheme (e.g., Genesis or Discovery Health). Each time a user goes to see a doctor, the clinic will make a claim to his/her medical scheme on his/her behalf.

Hints: underlined words are the names of the classes.

Requirements: There should be all the following links: association, composition, inheritance, multiplicity and navigation. The clinic class diagram should have at least 2 operations (those you consider are most important and express the relationship with other classes). [9]

Question 3: Interfaces [8 marks]

Given the following code

```
public class MyClass implements Cloneable, Comparable  
{  
    .....  
}
```

a) Explain what a **Cloneable** interface is and what it is used for. [2]

b) If there is no **clone()** method in the **MyClass** class, can we compile the program successfully and why? [2]

c) Explain what a **Comparable** interface is and what it is used for? [2]

d) If there is no **compareTo()** method in the **MyClass** class, can we compile the program successfully and why? [2]

Question 4: Generics [8 marks]

Given the following class

```
import java.util.*;
public class AList {
    private ArrayList ll;
    public AList() {
        ll = new ArrayList();
    }
    public void append(Object item) {
        ll.add(item);
    }
    public Object get(int index) {
        return ll.get(index);
    }
    public Object[] toArray() {
        return ll.toArray();
    }
}
```

- a) Explain what generic classes are and why they are needed. [2]

- b) Convert the **AList** class into a generic class called **GList**. [2]

- c) Given the following code using the **AList** and **GList** classes, indicate whether or not each statement is correct (with TRUE meaning that it is correct, and FALSE meaning it is not correct) . For each line, assume that all the ones prior to it are correct. [4]

True/False	Code
TRUE	<code>public static void main(String[] args) {</code>
	<code>AList al = new AList<String>();</code>
	<code>GList gl = new GList<String>();</code>
	<code>String str = "Hello CS1016";</code>
	<code>al.append(str);</code>
	<code>gl.append(str);</code>
	<code>String out_str1 = al.get(0);</code>
	<code>String out_str2 = gl.get(0);</code>
	<code>gl.append(new Double(20));</code>
	<code>}</code>

Section C: Linked structures

Question 5: Linear Linked Lists [5 marks]

- a) Describe the differences between a singly linked list and a doubly linked list. In particular what are the advantages and disadvantages of each? [2]

- b) Write a delete head node method **deleteHeadNode()** for the singly linked list given below. The method should remove the head node and return true if the list contains at least one node. Return false if the list is empty. [3]

```
1
2 public class LinkedList
3 {
4     private class Node
5     {
6         private String item;
7         private Node link;
8
9         public Node( )
10        {
11            item = null;
12            link = null;
13        }
14
15        public Node(String newItem, Node linkValue)
16        {
17            item = newItem;
18            link = linkValue;
19        }
20    } //End of Node inner class
21
22    private Node head;
23
24    public LinkedList( )
25    {
26        head = null;
27    }
28
```


Question 7: GUIs [12 Marks]

Answer the following questions with reference to the program below.

```
1  /** Menu Gui Answer **//
2  import javax.swing.JFrame;
3  import javax.swing.JPanel;
4  import java.awt.GridLayout;
5  import java.awt.Color;
6  import javax.swing.JMenu;
7  import javax.swing.JMenuItem;
8  import javax.swing.JMenuBar;
9  import java.awt.event.ActionListener;
10 import java.awt.event.ActionEvent;
11
12 public class GuiQuest extends JFrame implements ActionListener
13 {
14     public static final int WIDTH = 300;
15     public static final int HEIGHT = 200;
16
17     private JPanel redPanel;
18     private JPanel whitePanel;
19     private JPanel bluePanel;
20
21     public static void main(String[] args)
22     {
23         GuiQuest gui = new GuiQuest();
24         gui.setVisible(true);
25     }
26
27     public GuiQuest()
28     {
29         setSize(WIDTH, HEIGHT);
30         setTitle("Menu Demonstration");
31         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
32         setLayout(new GridLayout(1, 3));
33
34         redPanel = new JPanel();
35         redPanel.setBackground(Color.LIGHT_GRAY);
36         add(redPanel);
37
38         whitePanel = new JPanel();
39         whitePanel.setBackground(Color.LIGHT_GRAY);
40         add(whitePanel);
41
42         bluePanel = new JPanel();
43         bluePanel.setBackground(Color.LIGHT_GRAY);
44         add(bluePanel);
45
46         JMenu outerMenu = new JMenu("Actions");
47
48         JMenu colorMenu = new JMenu("Add Colors");
```

```

49
50     JMenuItem redChoice = new JMenuItem("Red");
51     redChoice.addActionListener(this);
52     colorMenu.add(redChoice);
53     JMenuItem whiteChoice = new JMenuItem("White");
54     whiteChoice.addActionListener(this);
55     colorMenu.add(whiteChoice);
56
57
58     JMenuItem blueChoice = new JMenuItem("Blue");
59     blueChoice.addActionListener(this);
60     colorMenu.add(blueChoice);
61
62     outerMenu.add(colorMenu);
63
64     JMenuBar bar = new JMenuBar();
65     bar.add(outerMenu);
66     setJMenuBar(bar);
67 }
68
69 public void actionPerformed(ActionEvent e)
70 {
71     String buttonString = e.getActionCommand();
72
73     if (buttonString.equals("Red"))
74         redPanel.setBackground(Color.RED);
75     else if (buttonString.equals("White"))
76         whitePanel.setBackground(Color.WHITE);
77     else if (buttonString.equals("Blue"))
78         bluePanel.setBackground(Color.BLUE);
79     else
80         System.out.println("Unexpected error.");
81 }
82 }
83

```

a) Illustrate the GUI produced by the above program when the program is run and the menu items selected. You should illustrate the complete frame. [4]

b) Explain the code at lines 21 – 25. Describe in your own words what it does. [2]

c) What changes are required so that the menu can include “exit” on the first level of the menu? In answering this question reference the code where changes must be added or changed. [4]

Line	New code:

Question 9: Trees [5 marks]

a) Explain a use of a binary tree. Illustrate your answer with a diagram. [3]

b) Provide the **algorithm** for a pre-order walk around a binary tree that prints out the value at each node. [2]

Section D: Ethics, Cyberlaw and Development

Question 10: Ethics, Cyberlaw and Development [15 marks]

a) Why is there a danger of a policy vacuum with respect to Computer Ethics? [5]

b) State the central idea of Utilitarianism in your own words. [4]
