FACULTY OF SCIENCE

ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	**COMPUTER SCIENCE 2A** CSC02A2
CAMPUS	AUCKLAND PARK CAMPUS (APK)
SUMMATIVE ASSESSMENT OPPORTUNITY III	SAO3
DATE: 2021-05-18	**SESSION:** 14:00 - 15:00
ASSESSOR(S):	**MR. A. MAGANLAL**
	MR. S. SITHUNGU
MODERATOR:	**MR. R. MALULEKA**
DURATION: 60 MINUTES	**MARKS:** 50

Please read the following instructions carefully:

1. You must complete the assessment **by yourself** within the prescribed time limits.
2. No communication concerning the assessment is permissible during the assessment session except with **ACSSE** staff members.
3. You are bound by all university regulations including, but not limited, to assessment, plagiarism, and ethical conduct.
4. You may not directly take any code from any source, including your own previous submissions. All code must be written by yourself during the assessment.
5. If you **do not have access to a computer** then you can do a *pen and paper submission*:
    - Write *cleanly* and *legibly*.
    - Make use of CamScanner app to create a PDF from your written work.
6. **All answers** must be in a *single* **PDF** file. Make sure your details appear at the top of the first page of the PDF file.
7. Complete the **Honesty Declaration: Online Assessment** and upload it as part of your submission. The completed **Honesty Declaration** is required for a submission to be eligible to be marked.
8. Your answers to the question (in a single PDF file) together with the declaration must be submitted in a zip archive named in the following format:
   `SURNAME_INITIALS_STUDENTNUMBER_CSC2A_2021_SAO3.zip`
9. Additional time for submission is allowed for as per the posted deadlines on EVE.
10. This paper contains **5** question(s).
11. This paper consists of **3** page(s) excluding the cover page.

QUESTION 1: Graphical User Interfaces

(a) **Discuss** how a complex **JavaFX** *graphical user interface* is ***created***. Your discussion must include the following: **[05]**

- The role/usage of **Pane Node**s.
- The role/usage of **Control Node**s.
- The relationship between **Pane Node**s and **Control Node**s.

(b) **Analyse** the following **Java** code segment and answer the questions that follow:

```
1 Button btn = new Button();
2 btn.setText("Say 'Hello World'");
3 btn.setOnAction(new EventHandler<ActionEvent >() {
4   @Override
5   public void handle(ActionEvent event) {
6     Sysout.out.println("Hello World!");
7   }
8 });
```

i. **Which** *signal* is being created? **[01]**

ii. **EventHandler** is an *interface*. On line 3 an instance of this interface is *created*. **Why** does this compile given the fact that instances of interface cannot be created? **[02]**

iii. **Discuss** the *relationship* between an ***Event*** and ***EventHandler***. **[02]**

Total: 10

QUESTION 2: Binary IO

(a) **When** would you choose to use a **PrintWiter** over an **ObjectOuputStream**? **[03]**

(b) **Which** *data members* of the following class will be ***written*** to file with an **ObjectOutput-Stream**? **Provide** a reason for your answer. **[02]**

```
1 public class StreamClass
2 {
3   public int            a;
4   public static int      b;
5   private transient int  c;
6 }
```

Total: 5

~~ Assessment continues on the next page. ~~

QUESTION 3: Advanced Java Programming

Analyse the following **Java** code segment and answer the questions that follow:

```
1  public class GenericBuilder<A, B> {
2    private A    basic;
3    private B[]  complex;
4    private static A CLASS_G_A;
5
6    public void createA()  {
7      basic = new A();
8    }
9
10   public void createB()  {
11     complex = (B[]) (new Object[5]);
12   }
13 }
```

The *code* shown above does not ***compile***. **Indicate** the line(s) where ***problem***(s) occur and **provide Java** source code that will correct the problem.

Total: 5

QUESTION 4: Design Patterns and UML

You need to connect to a remote machine using its **IPAddress**, but the connection library you are using has a class which contains a function that only accepts a **MACAddress**. Luckily, you learnt about the **Adapter Design Pattern** and you have a class that can convert an **IPAddress** into a **MACAddress**.

```
1  public class MACAddress { /* code omitted */ }
2  public class ConnectionInfo { /* code omitted */ }
3
4  public class IPAddress
5  {
6    public static MACAddress convert(IPAddress address) {/* code omitted */}
7  }
8
9  public class LegacyConnectionManager
10 {
11   public ConnectionInfo createConnection(MACAddress address) {/* code omitted */}
12 }
```

(a) **Provide** a UML class diagram for the **Adapter Design Pattern** applied to the problem stated above. **[10]**

(b) **Provide Java** source code for the concrete adapter class. **[05]**

Total: 15

~~ Assessment continues on the next page. ~~

QUESTION 5: Cold Code

Provide Java source code for the following problem. You can assume that all relevant packages have been imported.

Create a static ***getLowestCostedDevice*** method that will take a **File** handle as a parameter and return the cheapest **Device** found in the binary file using ***Automatic Resource Management***. The number of objects is written as **int** at the start of the file.

```
1  public class Device implements java.io.Serializable {
2    private String  deviceSerialNumber;
3    private String  deviceName;
4    private double  deviceCost;
5
6    public Device(String serialNumber, String name, double cost){ /*code omitted*/}
7
8    public String getDeviceSerialNumber()  {  return deviceSerialNumber; }
9    public String getDeviceName()  {  return deviceName; }
10   public double getDeviceCost()  {  return deviceCost; }
11 }
```

Total: 15

~~ THE END ~~