



## FACULTY OF SCIENCE

### ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	COMPUTER SCIENCE 2A CSC02A2
CAMPUS	AUCKLAND PARK CAMPUS (APK)
FSAO	JUNE
DATE: 2019-06-07	SESSION: 14:00 - 16:00
ASSESSOR(S):	MR. A. MAGANLAL MR. B. GREAVES
MODERATOR:	MR. R. MALULEKA
DURATION: 120 MINUTES	MARKS: 100

Please read the following instructions carefully:

1. Answer **all** the questions.
2. Answer questions in order.
3. Answer only in the examination books provided.
4. The use of calculators is *not* permitted.
5. Write ***cleanly and legibly***.
6. This paper contains **10** questions.
7. This paper consists of **3** pages excluding the cover page.

**QUESTION 1: Java Overview**

- (a) **Discuss** the *differences* between **compiled programming languages** and **byte code interpreted programming languages**. [04]
- (b) **Name** two (2) **design goals** of the **Java** language. [02]
- (c) **Discuss** the *differences* between C++ and Java with regards to **multi-threading**. [04]

Total: 10

**QUESTION 2: Elementary Java Programming**

- (a) **Name** and **describe** the two (2) *categories* of **data types** available in **Java**. **Provide** an **example** for each. [06]
- (b) **List** two (2) *advantages* of **using methods** in a program as opposed to having code defined in the **main** method. [04]

Total: 10

**QUESTION 3: Text Processing and Persistence**

- (a) When writing objects using an **ObjectOutputStream** some data members cannot be written. **Which** *data members* cannot be **serialized**? [02]
- (b) **Discuss** the *role* of **BufferedOutputStreams** when writing a binary file. [02]
- (c) **What** does the **isDirectory()** method in the **File** class return? [01]
- (d) **Provide** a *single regular expression* that matches *all load shedding schedules* in the following format. [05]
  - 6A Dressdorp 14:00
  - 8Z Chessburg 20:00
  - 1B Jeffstown 06:30
  - 9M Bergplein 12:30

Total: 10

**QUESTION 4: Object Orientation**

- (a) **Name** and **describe** two (2) *principles of Object Oriented design*. [06]
- (b) **Define** the *concept* of **overshadowing** in classes. [02]
- (c) **Provide** a *definition* of a **marker interface**. [02]

Total: 10

**QUESTION 5: Graphical User Interfaces**

- (a) **Discuss** the **Swing** graphical user interface *framework*. Your discussion must include how this framework works. [04]
- (b) **Name** and **discuss** any two (2) **layout managers** found in **JavaFX**. [04]
- (c) **Name** any two (2) **helper classes** in the **JavaFX**. [02]

Total: 10

~~ Assessment continues on the next page. ~~

**QUESTION 6: Advanced Java Programming**

- (a) **List** two (2) *restrictions* that apply to **generics** in Java. [02]
- (b) **Define** *type erasure*. [03]
- (c) With regards to **Java multi-threaded programming** **provide** a definition of a **task** and **thread**, and **discuss** how these two concepts are **related**. [05]

**Total: 10****QUESTION 7: Design Patterns**

- (a) **Discuss** two (2) *limitations* of the **Object Pool Design Pattern**. [04]
- (b) **Name** two (2) **behavioural design patterns**. [02]
- (c) **Name** two (2) of the **golden rules** for *design patterns*. [04]

**Total: 10****QUESTION 8: UML**

**Provide** a UML class diagram of the **Proxy Design Pattern** applied to the following problem.

You have been given access to a class called **RemoteReader**. This class represents functionality that is called on-demand to read information from a remote service provider but should not be freely accessible. The client application requires a fix for this issue.

```
1 public class RemoteReader
2 {
3     public String remoteRead(String webAddress)
4     {
5         /* code omitted to read information from the web address*/
6     }
7 }
```

**Total: 10****QUESTION 9: Cold Code**

Provide **Java** source code for the following problem. You can assume that all relevant packages have been imported. A **Student** has the following information:

- STU\_NUM - Integer student number
- STU\_NAME - String name of the student
- STU\_AGE - Integer age of the student

Create a **writeStudent** method that will write the above **Student** to the binary file *studentInfo.dat*. The Student's information is passed as three (3) parameters to the **writeStudent** method.

**Total: 10**

~~ Assessment continues on the next page. ~~

**QUESTION 10: Fill-in Code**

Read the code below and fill in **the missing code** (in segments labelled as A to J). Write down the letter and the correct answer next to it. E.g. (K) Kiwi().

```

1  public class Shape{ //Base class
2      private double x, y;
3      public double getX(){ return x; }
4      public double getY(){ return y; }
5  }
6  public class Circle _(A(1 Mark))_ Shape{ //Derived class
7      private int diameter;
8      public int getDiameter(){ return diameter; }
9  }
10 public class Square _(A(1 Mark))_ Shape{ //Derived class
11     private int size;
12     public int getSize(){ return size; }
13 }
14 public class ShapeCanvas extends Canvas{ //Specialized JavaFX Canvas
15     private ArrayList<Shape> shapes = null;
16     public void setShapes(ArrayList<Shape> shapes){
17         _(B(1 Mark))_.shapes = shapes;
18     }
19     public void redrawCanvas(){
20         GraphicsContext gc = _(C(1 Mark))_;
21         _(D(1 Mark))_.setFill(Color.BLUE);
22         _(E(1 Mark))_(shapes != null){ //Ensure that shapes isn't null
23             for(Shape shape: shapes){
24                 if(shape _(F(1 Mark))_ Circle){ //Test if shape is a Circle
25                     Circle c = _(G(1 Mark))_shape; //Cast to derived class
26                     gc. _(H(1 Mark))_(c.getX(), c.getY(), c.getDiameter(),
27                         ↪ c.getDiameter()); //Draw Circle
28                 }else if(shape _(I(1 Mark))_ Square){ //Test if shape is a Square
29                     Square s = /* omitted */shape; //Cast to derived class
30                     gc. _(J(1 Mark))_(s.getX(), s.getY(), s. _(K(1 Mark))_, s. _(L(1
31                         ↪ Mark))_); //Draw Square
32                 }
33             }
34         }
35     }
36 }

```

Total: 10

~~ THE END ~~