

UNIVERSITY OF SWAZILAND

DEPARTMENT OF COMPUTER SCIENCE

CSC242 — OBJECT ORIENTED PROGRAMMING

RE-SIT EXAMINATION

JULY 2018

Instructions

1. The time allowed is **THREE (3) HOURS**.
2. Read all the questions in **Section A** and **Section B** before you start answering any question.
3. Answer **all** questions in Section A. Answer **any two** questions of Section B. Maximum mark is 100.
4. Use correct notation and show all your work on the answer script.

**DO NOT OPEN THIS PAPER UNTIL YOU ARE INSTRUCTED
TO DO SO BY THE INVIGILATOR**

Section A

Answer all questions in this section. Questions 1 and 2 are based on the following case description.

You have been asked to write a program to be used by your small community library at Emvakwelitje in Mbabane. The library has books, videos and CDs that it loans to its users. All library materials have a unique identification number and a title. In addition, books have one or more authors, videos have one producer and one or more actors, while CDs have one or more entertainers. The library maintains one or more copies of each library item (book, video or CD).

Copies of all library materials can be loaned to users. For every loan, the library records the user, the loan date and time, the return date and time. For users, the library maintains their name, address and phone number.

Question 1 [25]

Draw a class diagram for the description above. Your diagram must show all attributes, and associations, where appropriate. [25]

Question 2 [25]

- a Add design attributes to capture the associations shown in the diagram obtained in Question 1 above. [3]
- b Using C++ notation, write suitable class definitions for your library system. All constructors must be defined, but for other member functions, only write the prototypes. [22]

Section B

Question 3 [25]

- a Explain the following concepts. Give examples where possible? [5]
 - (a) Function Prototype
 - (b) Function Signature
 - (c) Function Overloading
 - (d) Friend Function
 - (e) Access member function
- b Constructor:
 - (a) What is a constructor function? [2]
 - (b) How and when is a constructor function called/used? [2]
 - (c) A class may have more than one constructor function. How is this possible? [2]
- c Templates and Polymorphism
 - (a) What is a polymorphic function? [2]
 - (b) Explain two (2) cases (concepts) which give rise to polymorphic functions. [2]
 - (c) What is function template? [2]
 - (d) Do you see any relationship between function templates and polymorphism? [2]
- d Classes and Objects
 - (a) What is the relationship between a class and an object? [2]
 - (b) What is difference between private and protected members of a class? [2]
 - (c) What is the difference between a private and a public base class? [2]

Question 4 [25]

- a List the three situations whereby a copy constructor automatically executes. [3]
- b Briefly discuss an abstract class, virtual functions and pure virtual functions. Use a suitable example class definition to demonstrate the above concepts. [8]
- c Write down a statement that declares a dynamic two-dimension array of **double** type. The number of rows must be 10 while columns must be 15. [2]

- d Write down lines of code that initialize the two dimension array declared in part (c) above. All array entry values must be initialized to 0.1 [5]
- e Write down lines of code that deallocate memory for the 2-dimensional array initialized above. [3]
- f Using templates, write a function that returns the smallest of three argument values. [4]

Question 5 [25]

- a Mark the following statement as true or false about **struct**. [6]
 - (a) All members of a **struct** must be of different types.
 - (b) A **struct** is a definition not a declaration.
 - (c) A **struct** variable must be declared after a struct definition
 - (d) A function cannot return a value of the type **struct**.
 - (e) A **struct** member is accessed by using the operator `.`
 - (f) A member of a **struct** can be another **struct**.
- b What is a dangling pointer? [2]
- c Explain the following statement. [2]


```
using namespace std;
```
- d Consider the following declaration, and answer the following question. [3]


```
enum fruitType {ORANGE, APPLE, BANANA, GRAPE, STRAWBERRY,
                MANGO, GUAVA, PINEAPPLE, KIWI};
fruitType fruit;
```

 - (a) What is the value of `static_cast<int>(STRAWBERRY)`
 - (b) What is the value, if any, of the following expression?
`static_cast<fruitType>(static_cast<int>(MANGO) - 2)`
 - (c) What is the value, if any, of the expression: `BANANA <= KIWI`
- e Explain three (3) exception handling techniques. [6]
- f Briefly discuss the recursion problem solving approach. [6]