

UNIVERSITY OF SWAZILAND
DEPARTMENT OF COMPUTER SCIENCE
CSC242 — OBJECT ORIENTED PROGRAMMING
MAIN EXAMINATION

MAY 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

Question 1 [25]

- i Write down the syntax of a *class* definition? [3]
- ii What is the default access specifier of a *structure* type? [1]
- iii The *private* members of a class are inaccessible outside the class scope. Can they be manipulated outside the class? If so, how? If not, why?[2]
- iv Explain a *const* function. [2]
- v What is ADT and what does ADT stand for? How does O-O programming handle ADT? [4]
- vi Name and explain three basic concepts of object oriented programming (OOP) [6].
- vii Describe the name. destructor and explain how and when destructors work in a program. [6]
- viii What is a *postcondition* of a function? [1]

Question 2 [25]

- a Define the **class** *pointType* to implement the properties of a point in two-dimensional plane. [6]

Your class must contain functions to individually set and retrieve x and y coordinates, constructors to initialize *pointType* objects and find distance between this point and another point.

If $p1 = (x_1, y_1)$ and $p2 = (x_2, y_2)$ are *pointType* objects, then $p1.distance(p2)$ returns the distance between $p1$ and $p2$. The formula for distance is given as; $Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

- b Write the definitions of the member functions of the **class** *pointType* defined above in part a. [6]
- c If we know two points on a line, we can describe various properties of a line, such as whether the line is vertical, horizontal, or slanted, and if the line is nonvertical, then find its slope. [6]

Define the **class** *lineType* with two private data members of *pointType* to store the coordinates of two points on a line in a two-dimensional plane.

Among others your class must contain functions to determine if the line is vertical, horizontal, or slanted, find the slope of the line, \mathbf{m} , and output the equation of the line in slope interception form, $y = \mathbf{m}x + c$. The equation of a slope, \mathbf{m} is $\mathbf{m} = \frac{y_2 - y_1}{x_2 - x_1}$. If the line is vertical, then output the equation in the form $x = a$ where a is a real number.

- d Write the definitions of member function of **class** *lineType* defined in part c.[6]
- e What kind of a relationship does **class** *pointType* and **class** *lineType* have? [1]

Section B

Question 3 [25]

- a Mark the following statements as true or false. [5]
 - i In C++ pointer variables are declared using the word `pointer`.
 - ii In C++, the dereferencing operator has a higher precedence than the dot operator.
 - iii The statement `delete p`; deallocates the variable pointer `p`.
 - iv if a pointer `p` points to a dynamic array, the elements of `p` can be processed using a range-based `for` loop.
 - v The binding of virtual functions occurs at compile time.
- b Name any C++ built operator on classes. [1]
- c Explain the difference between *deep copy* and *shallow copy* of data. [2]
- d With suitable examples, explain each of the following terms: static binding and dynamic binding. [4]
- e With a suitable example, explain how abstract classes are defined. [2]
- f Answer the following questions relating to two-dimension dynamic pointer variable `votes`.
 - i Write a statement that declares `votes` to be a pointer of type `int`. [1]
 - ii Write C++ code that dynamically creates a two-dimensional array of 50 rows and 10 columns and `votes` contains the base address of that array. [4]
 - iii Write a C++ code that outputs the data stored into the array `votes` one row per column. [3]
 - iv Write a C++ code that deallocates the memory space occupied by the two-dimensional array to which `votes` points. [3]

Question 4 [25]

- a Write down the syntax for class template declaration. [1]
- b Consider the following declaration:

```
class stranger
{
    ...
};
```

- i Write a statement that shows the declaration in the **class** *stranger* to overload the **operator =**. [2]
- ii Write a statement that shows the declaration in the **class** *stranger* to overload the binary **operator -** as a **friend**. [2]
- iii Write a statement that shows the declaration in the **class** *stranger* to overload the the post increment operator **++** as a member function. [2]
- c Consider the following **class** *arrayClass* declaration, and the definition of the assignment operator overloading member function.

```
class arrayClass
{
private:
    int *list;
    int length;
    int maxSize;
private:
    const arrayClass& operator=(const arrayClass& otherList) //
Line 1
    {
    if (this != &otherList) // Line 2
    {
        delete [] list; // Line 3
        maxSize = otherList.maxSize; // Line 4
        length = otherList.length; // Line 5
```

```

        list = new int[maxSize]; // Line 6
        for(int i = 0; i < length; i++) // Line 7
            list[i] = other.list[i]; // Line 8
    }
    return *this; // Line 9
};

```

Explain each statement of the definition of **operator=** that have a comment *Line N*, $N = 1 \dots 9$. Where necessary discuss the role/importance of having these statements. [12]

d) Consider the following declaration:

```

template <class type> class stranger
{
    ...
private:
    type a;
    type b;
};

```

- i) Write a statement that declares *sObj* to be an object of type *strange* such that the **private** member variables *a* and *b* are of type **int**. [2]
- ii) Write a statement that shows the declaration in the **class** *strange* to overload the operator **==** for the **class** *strange* as a member function. [2]
- iii) Assume two objects of type *strange* are equal if their corresponding member variables are equal. Write the definition of the function operator **==** for the **class** *stranger*, which is overloaded as a member function. [2]

Question 5 [25]

- a Give two reasons why we place the class definition file in the header file, and the definition of member functions in the implementation file? [2]
- b For class templates, why does the mechanism of separation of header files and implementation file not work? [2]
- c Explain why you would need **public**, **protected**, and **private** members in a class [3]
- d When initializing an object of a derived class, which constructor executes first between that of the base class or derived class? [1]
- e When exiting the scope of an object of a derived class, which destructor executes first between that of the base class or derived class? [1]
- f Explain the following function prototype, for a method in a **class** called **Object**. [3]
Object* Comp(const** *Object* & obj) **const**;
- g When working with multiple inclusion of header files, especially inheritance, compile-time errors such as, "an identifier has already been declared" occur.

With an aid of code snippet, explain how to avoid this problem. [4]

- h With respect to exception handling, answer the following questions.
 - i What will happen if an exception is thrown but not caught? [2]
 - ii How many parameters can a **catch** block have? [1]
 - iii Which **catch** block catches all types of exceptions? [1]
- i Consider the following recursive function.

```
int recTes(int a, int b)
{
    if (a < b)
        return a + b;
```

```
    else
      return recTes(a - b, b+1);
}
```

- (a) identify the base case. [1]
- (b) identify the general case. [1]
- (c) what is the output of the following statement? [3]
`cout << recTes(-10, 8) << endl;`