

University of Eswatini

Department of Computer Science

Final Examination

2018/19

Title of Paper: Object Oriented Programming

Course Number: CSC242

Time Allowed: Three (3) hours

Instructions: Answer **ALL** questions

You are not allowed to open this paper until you have been told to do so by the invigilator.

Part I – OPP Basic Concepts

QUESTION ONE

{15 marks}

Discuss the following concepts using examples:

- 1.1) Abstract class {3 marks}
- 1.2) Polymorphism {3 marks}
- 1.3) Inheritance {3 marks}
- 1.4) Encapsulation {3 marks}
- 1.5) Polymorphism {3 marks}
- 1.6) Interface {3 marks}

Part II – True or false

QUESTION TWO

{15marks}

Indicate whether the statement is true or false.

- 2.1) In C++, class is a reserved word and it defines only a data type. {1 mark}
- 2.2) If the heading of a member function of a class ends with the word const, then the function member cannot modify the private member variables, but it can modify the public member variables. {1 mark}
- 2.3) Given the declaration

```
class myClass
{
public:
    void print(); //Output the value of x;
    MyClass();
```

```
private:  
    int x;  
};
```

```
myClass myObject;
```

The following statement is legal.

```
myObject.x = 10;
```

{1 mark}

2.4) If an object is created in a user program, then the object can access both the public and private members of the class.

{1 mark}

2.5) In multiple inheritance, the derived class has more than one base class.

{1 mark}

2.6) In protected inheritance, public and protected members of the base class become the protected members of the derived class.

{1 mark}

2.7) The dereferencing operator is also known as the indirection operator and refers to the object to which its operand points.

{1 mark}

2.8) In C++, the dot operator has a lower precedence than the dereferencing operator.

{1 mark}

2.9) In C++, the member access operator arrow is >>.

{1 mark}

2.10) Given the declaration

```
int *p;
```

The statement

```
p = new int[50];
```

dynamically allocates an array of 50 components of type int and p contains the base address of the array.

{1 mark}

2.11) A memory leak is an unused memory space that cannot be allocated.

{1 mark}

2.12) When writing the definition of a friend function, the name of the class and the scope resolution operator precede the name of the friend function in the function heading.

{1 mark}

2.13) The associativity of the operator = is from right to left.

{1 mark}

2.14) An object that is being thrown cannot be an anonymous object.

{1 mark}

2.15) The order of the catch blocks does not affect the program.

{1 mark}

Part III – Multiple Choice

QUESTION THREE

{15 marks}

Identify the choice that best completes the statement or answers the question.

3.1) Which of the following class definitions is correct in C++?

a. class studentType

```
{
    public:
        void setData(string, double, int);
    private:
        string name;
};
```

b. class studentType

```
{
    public:
        void setData(string, double, int);
        void print() const;
    private:
        string name;
        double gpa;
}
```

c. class studentType

```
{
    public void setData(string, double, int);
    private string name;
};
```

```

d. studentType class
{
    public: void setData(string, double, int);
    private: string name;
};

```

Figure 1:

clockType
-hr: int
-min: int
-sec: int
+setTime(int, int, int): void
+getTime(int&, int&, int&) const: void
+printTime() const: void
+incrementSeconds(): int
+incrementMinutes(): int
+incrementHours(): int
+equalTime(const clockType&) const: bool

{1 mark}

3.2) The word _____ at the end of the member functions in the accompanying class clockType in Figure 1 specifies that these functions cannot modify the member variables of a clockType object.

- a. static
- b. const
- c. automatic
- d. private

{1 mark}

3.3) In C++, you can pass a variable by reference and still prevent the function from changing its value by using the keyword _____ in the formal parameter declaration.

- a. automatic
- b. private
- c. static
- d. const

Figure 3:

```

class secretType

```

```

{
public:
    static int count;
    static int z;

    secretType();
    secretType(int a);
    void print();
    static void incrementY();

private:
    int x;
    static int y;
};

secretType::secretType()
{
    x = 1;
}
secretType::secretType(int a)
{
    x = a;
}
void secretType::print()
{
    cout << "x = " << x << ", y = " << y
        << "z = " << z
        << ", count = " << count << endl;
}
static void secretType::incrementY()
{
    y++;
}

```

{1 mark}

3.4) Consider the accompanying class and member functions definitions in Figure 3. How many constructors are present in the class definition above?

- a. none
- b. one
- c. two
- d. three

{1 mark}

3.5) Consider the following class definition:

```
class dClass: bClass
```

```
{  
    //class members list  
};
```

The class dClass is derived from the class bClass using the ____ type of inheritance.

- a. public
- b. private
- c. protected
- d. static

{1 mark}

3.6) ____ is the ability to use the same expression to denote different operations.

- a. Inheritance
- b. Encapsulation
- c. Polymorphism
- d. Composition

{1 mark}

3.7) The ____ members of an object form its external state.

- a. private
- b. public
- c. protected
- d. static

{1 mark}

3.8) What is the output of the following code?

```
int *p;  
int x;  
x = 12;  
p = &x;  
cout << x << ", ";  
*p = 81;  
cout << *p << endl;
```

- a. 12, 12
- b. 12, 81
- c. 81, 12
- d. 81, 81

{1 mark}

3.9) The name of the function to overload the operator <= is ____.

- a. overload<=
- b. <=new
- c. operator<=
- d. <=operator

{1 mark}

3.10) A catch block can have, at most, ____ catch block parameter(s).

- a. zero
- b. one
- c. two
- d. three

{1 mark}

that the initial balance was invalid. Provide three member functions. Member function **credit** should add an amount to the current balance. Member function **debit** should withdraw money from the **Account** and ensure that the debit amount does not exceed the Account's balance. If it does, the balance should be left unchanged and the function should print a message indicating "Debit amount exceeded account balance." Member function **getBalance** should return the current balance. Create a program that creates two **Account** objects and tests the member functions of class **Account**.

4.2) Create a UML diagram for this class (Square) {7 marks}

QUESTION FIVE

{13 marks}

5.1) Illustrate how to subtract two complex numbers using operator overloading of the - operator. {7 marks}

5.2) Program to display largest among two numbers using function templates. {6 marks}

QUESTION SIX

{10 marks}

What is the output of the following code?

```
6.1) #include <iostream>
      using namespace std;

      void f1(int x, int &y, int *z)
      {
        x++;
        y++;
        (*z)++;
      }

      int main()
      {
        int i = 1, j = 1, k = 1;
        f1(i, j, &k);

        cout << "i is " << i << endl;
```

```
cout << "j is " << j << endl;
cout << "k is " << k << endl;
```

```
return 0;
}
```

{2 marks}

6.2)

```
#include <iostream>
#include <fstream>
using namespace std;
```

```
int main()
```

```
{
    ofstream output;
```

```
    // Create a file
    output.open("scores.txt");
```

```
    // Write two lines
    output << "John" << " " << "T" << " " << "Smith"
        << " " << 90 << endl;
    output << "Eric" << " " << "K" << " " << "Jones"
        << " " << 85;
```

```
    output.close();
```

```
    ifstream input;
```

```
    // Open a file
    input.open("scores.txt");
```

```
    // Read data
    char firstName[80];
    char mi;
    char lastName[80];
    int score;
    input >> firstName >> mi >> lastName >> score;
    double sum = score;
```

```
    input >> firstName >> mi >> lastName >> score;
    sum += score;
```

```
    cout << "Total score is " << sum << endl;
```

```
input.close();
```

```
return 0;
```

```
}
```

{2 marks}

6.3) Show the output of the following program:

If you enter 1 0, what is the output of the following code?

```
#include <iostream>
using namespace std;
```

```
int main()
```

```
{
```

```
    // Read two integers
```

```
    cout << "Enter two integers: ";
```

```
    int number1, number2;
```

```
    cin >> number1 >> number2;
```

```
    try
```

```
    {
```

```
        if (number2 == 0)
```

```
            throw number1;
```

```
        cout << number1 << " / " << number2 << " is "
             << (number1 / number2) << endl;
```

```
    }
```

```
    catch (int e)
```

```
    {
```

```
        cout << "Exception: an integer " << e <<
```

```
             " cannot be divided by zero" << endl;
```

```
    }
```

```
    cout << "Execution continues ..." << endl;
```

```
    return 0;
```

```
}
```

{2 marks}

6.4)

```
#include <iostream>
```

```
using namespace std;
```

```
class Parent
{
public:
    virtual void f()
    {
        cout << "invoke f from Parent" << endl;
    }
};
```

```
class Child: public Parent
{
public:
    void f()
    {
        cout << "invoke f from Child" << endl;
    }
};
```

```
void p(Parent a)
{
    a.f();
}
```

```
int main()
{
    Parent a;
    a.f();
    p(a);

    Child b;
    b.f();
    p(b);

    return 0;
}
```

{2 marks}

6.5)

```
#include <iostream>
using namespace std;

class Parent
```

```
{
public:
    Parent()
    {
        cout << "Parent's no-arg constructor is invoked" << endl;
    }

    ~Parent()
    {
        cout << "Parent's destructor is invoked" << endl;
    }
};

class Child: public Parent
{
public:
    Child()
    {
        cout << "Child's no-arg constructor is invoked" << endl;
    }

    ~Child()
    {
        cout << "Child's destructor is invoked" << endl;
    }
};

int main()
{
    Child c1;
    Child c2;

    return 0;
}
```

{2 marks}