University of Colombo, Sri Lanka

University of Colombo School of Computing

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

First Year Examination — Semester II– 2020/2021

SCS1209 — Object Oriented Programming - Part B

(Two (2) Hours for both part A & part B)

Answer ALL questions

Number of Pages = 8 Number of Questions = 2

To be completed by the candidate

Index Number | | | | | | | | |

Important Instructions to candidates:

- The medium of instruction and questions is **English**.
- **Write your answers in English.**
- Note that questions appear on both sides of the paper. If a page is not printed, please inform the supervisor immediately.
- Write your index number on each and every page of the question paper.
- The duration of the paper is **Two (2)** hours for **both parts A & B**.
- This paper, **part B** has **2** questions on **8** pages.
- Answer **all** the questions in this **part B**
- Each question carries exactly **25 marks**.
- **Write your answers on the space provided** on this question paper.
- Any electronic device capable of storing and retrieving text including electronic dictionaries and mobile phones are **not allowed**.

To be completed by the examiners

1	
2	
Total	

Index Number | | | | | | | | |

1. (a). Only ONE answer is correct in the following 12 MCQs. Underline the **best suite answer** among the given options.

[2 x 12 marks]

i. Which of the following set of classes can be derived classes of the base class `Vehicle`

A. Train, Engine
B. Car, Truck
C. Engine, Wheel
D. Car, Wheel
E. Cart, Horse

ii. Which of the following is TRUE according to the following class definition?

```
class Student : private Person {
   // body of the class
};
```

A. Both public and protected data in `Person` will become private to `Student`
B. All public, protected and private data in `Person` will become private to `Student`
C. All public and protected data to `Student` will become private in `Person`
D. Only private data in `Person` will become private to `Student`
E. `Student` cannot access any data in `Person`

iii. Consider the fallowing 3 statements regarding inheriting constructors.

A. Derived class will not inherit the constructors from the base class.
B. Derived class constructors can call base class constructors.
C. When an instance of a derived class comes into existence, only the derived class constructor is automatically invoked.

Which of the above statements are TRUE?

A. A only.
B. B only.
C. C only.
D. A and B only.
E. A, B and C.

Index Number | | | | | | | | |

iv. Consider the fallowing 3 statements regarding the `Diamond Problem`.

A. It can be occurred when a class have two base classes.
B. It occurs in the classes which have a common base class.
C. You always get a complication error when it occurred.

Which of the above statements are TRUE?

A. A only.
B. B only.
C. A and B only.
D. B and C only.
E. A, B and C.

v. Consider the following 3 statements regarding the `Virtual Functions`?

A. It is a member function declared within the derived class.
B. It is mandatory to override the virtual functions.
C. There is always an additional performance penalty when using virtual functions.

Which of the above statements are TRUE?

A. A only.
B. B only.
C. C only.
D. A and C only.
E. A, B and C.

vi. The following shows one of the signatures of the function named draw() in the class named `Shape`.

```
void draw(int w) {}
```

Which of the following can NOT be considered as a valid signature to overload this function?

A. `void draw(char w)`
B. `void draw(double w)`
C. `void draw(int w, int h)`
D. `void draw(int w, char h)`
E. `int draw(int w)`

vii. Following is the definition of a function in the class named ClassA.

```
virtual void func(int x=0, int y=0) = 0;
```

Which of the following is FALSE regarding the above code?

A. This fuention returns 0 when it calls.
B. You can not create objects from ClassA.
C. You will get a compiler error if you remove the keyword `virtual`
D. It is not mandatory to pass 2 integers when calling this function.
E. ClassA is an Abstract class.

Index Number | | | | | | | | |

viii. Consider the fallowing 3 statements regarding the `Templates.` in C++

A. It is a feature of C++ that allows us to write one code for different data types.
B. We cannot use templates for user defined types.
C. Template is an example of compile time polymorphism.

Which of the above statements are TRUE?

A. A only.
B. C only.
C. A and B only.
D. A and C only.
E. A, B and C.

ix. What is the outcome of the following program written in C++?

```
template <int i>
void fun() {
   i = 20;
   cout << i;
}

int main() {
   fun<10>();
}
```

A. 10
B. 20
C. 30
D. Runtime Error
E. Compilation Error

x. Which of the following is TRUE concerning the logical errors of a program?

A. Logical errors can be tracked down easily.
B. Logical errors raise at runtime.
C. Logical error occurs because of incorrect syntax of your code.
D. Logical errors do not harm for the outcome of the program.
E. Your code doesn't compile if you have a logical error in your program.

xi. What type of errors can handle through exception handling?

A. Resource errors
B. Interface errors
C. Syntax errors
D. Compilation errors
E. Run time errors

Index Number | | | | | | | | |

xii. What is the output of the following program written in C++?

```
int main() {
   int x = 0;

   try {
        throw x;
        cout << "A ";
    }

   catch (int y) {
     cout << "B ";
   }

   catch(...) {
    cout<<"C ";
   }

   cout << "D ";
   return 0;
}
```

A. A B D
B. A C D
C. B D
D. B C D
E. A B C D

(b). What is the output of the following program written in C++?
(Assume that all the required lines in the header are already there in the code.)

```
class Base {};
class Derived: public Base {};

int main() {
   Derived d;
   try {
       throw d;
   }
    catch(Base b) {
        cout<<"Caught Base Exception";
   }
    catch(Derived d) {
        cout<<"Caught Derived Exception";
   }
   return 0;
}
```

Caught Base Exception

. . . . . .

[1 marks]

Index Number | | | | | | | | |

2. (a). Write the outputs of the following piece of programs written in C++.
(Assume that all the required lines in the header are already there in the code.)

i. .

```
class ClassA {
    public:
        ClassA() {cout<<"A ";}
        ~ClassA() {cout<<"~A ";}
};

class ClassB {
    public:
        ClassB() {cout<<"B ";}
        ~ClassB() {cout<<"~B ";}
};

class ClassC : public ClassA, ClassB {
    public:
        ClassC() {cout<<"C ";}
        ~ClassC() {cout<<"~C ";}
};

int main() {
    ClassC c;
}
```

A B C -C -B -A

[6 marks]

ii. (Hint: "\t" is to print the tab)

```
class ClassA {
    public: int a;
};
class ClassB: virtual public ClassA {
    public: int b;
};
class ClassC: virtual public ClassA {
    public: int c;
};
class ClassD: public ClassB, public ClassC {
    public: int d;
};
int main() {
    ClassD obj;
    obj.a = 5; obj.a = 10;
    obj.b = 10; obj.c = 20;
    cout<<" A in B: "<<obj.a;
    cout<<"\t A in C: "<<obj.a;
    cout<<"\n B: "<<obj.b;
    cout<<"\t C: "<<obj.c;
}
```

[4 marks]

Index Number | | | | | | | | |

(b). Define `Function Overloading` and `Function Overriding` by giving examples for each.

Function Overloading:

Function Overriding:

[4 marks]

(c). Output of the following program (written in C++) is given below;

Program:

```
int main()  {
    cout << ABC<int, char>(2, 'A') << endl;
    cout << ABC<char, int>('A', 2) << endl;
    cout << ABC<double, int>(2.5, 5) << endl;
}
```

output:

```
130
é
12.5
```

Implement the template named `ABC` to cater to the above requirement.

[6 marks]

Index Number | | | | | | | | |

(d). The function named `func(int )` is in the following program should return an exception when it gets a positive value and should return the same when it gets a negative value. Hence, it prints `A` when `y=65` and will print `-10`, when `y=-10`

```
int main() {
    int y = 65;
 try {
      cout<<func(y);
    }
    catch(char ch) {
        cout << ch;
    }
}
```

Implement the `func(int )` to cater to the above requirement.

[5 marks]

_____ ********** _____