



EASTERN UNIVERSITY, SRI LANKA
DEPARTMENT OF MATHEMATICS

FIRST YEAR EXAMINATION IN SCIENCE - 2015/2016

SECOND SEMESTER (MAY/JUNE, 2018)

CS 104 - OBJECT ORIENTED PROGRAMMING TECHNIQUES

Answer all questions

Time allowed: Two Hours

Q1. C++ is called object oriented programming (OOP) language because C++ language views a problem in terms of objects involved rather than the procedure for doing it.

- Briefly explain the concept of *Object Oriented Programming*. [15 Marks]
- Compare and contrast *Procedure Oriented Programming* and *Object Oriented Programming*. [25 Marks]
- What is a class diagram in *Unified Modeling Language (UML)*? [10 Marks]
- Describe the use of *scope resolution operator (::)*. [10 Marks]
- Write a C++ program using scope resolution operator that has a class named *Student* with data members *name*, *mark1* and *mark2*. Create functions *getInfo()* that asks the user to enter name and marks, *marksAverage()* that returns average of mark1 and mark2, and *printInfo()* that displays the name, marks and average marks on screen. [40 Marks]

Q2. An object is a software bundle of related variables and methods which are represents an entity in the real world that can be distinctly identified.

- Explain the difference between the following pair of terms: [30 Marks]
 - Member function* and *Friend function*,
 - Accessor* and *Mutator*,
 - Constructor* and *Destructor*.

b. Give the C++ syntax for the following definitions:

[15 Marks]

- i. A constructor,
- ii. A function,
- iii. A class.

c. What is *this* pointer in C++? Explain the situation where *this* pointer is used with the aid of a program.

[25 Marks]

d. Fix the compilation errors and predict the output of the following program.

[30 Marks]

```
class Test {
    private
        int x;
        int y;
        Test (int x = 0, int y = 0) {
            this.x = x;
            this.y = y;
        }
        void change(Test t) {
            this= t;
        }
        Test setX(a) {
            x =a;
            return *this;
        }
        Test setY(int b) {
            y = b;
            return this;
        }
        void print() {
            cout <<"x = " <<x <<"y = " <<y <<endl;
        }
}

int main(){
    Test obj1(15,20), obj2(12,34);
    obj1.setX(10).setY(20);
    obj1.print();
    obj2.print();
}
```

Q3. All object-oriented programming languages provide mechanisms that help you to implement the object oriented model.

- a. Briefly explain the concept of *inheritance* and its types with the aid of diagrams. [30 Marks]
- b. List the *access specifiers* used in C++. In a base class specifier of a derived class declaration, define the accessibility of inherited members. [15 Marks]
- c. Describe the problem with replicate the class, when you don't use virtual class. Explain how could fix the problem using virtual class with a suitable example program. [20 Marks]
- d. Answer the questions i. to v. based on the following C++ code: [35 Marks]

```
class Book {
    char name[20];
protected:
    int bookno;
public:
    void getdata();
    void displaydata();
};

class Author {
    char author[20], publisher[20];
public:
    void getdata();
    void showdata();
};

class Publication: public Book, public Author {
    int no_pages, year;
public:
    void get();
    void display();
};
```

- i. Which type of inheritance is illustrated in the above C++ code?
- ii. Write the names of all data members and member functions, which is/are accessible from objects of classes *Author* and *Publication*.

- iii. Rewrite the member function *get()* in class *Publication*, which calls the member function *getdata()* from its base classes.
- iv. Rewrite the member function *display()* in class *Publication*, which calls the member functions *displaydata()* and *showdata()* from its base classes.
- v. Write the *main* method to create an object for the class *Publication* and invoke the member functions of the class *Publication*.

Q4. OOP is about developing an application around its data, i.e., objects which provides the access to their properties and the possible operations in their own way.

- a. State clearly what is meant by *Polymorphism*. Define how the types of polymorphism are achieved. [20 Marks]
- b. Explain three differences between *overloading* and *overriding* in C++. [20 Marks]
- c. Write code fragments to illustrate the following concepts: [30 Marks]
 - i. Unary operator overloading,
 - ii. Binary operator overloading,
 - iii. Function with default arguments,
 - iv. Function with different types of arguments.
- d. Write a C++ program for a class *IncomeTax* with the following instructions: [30 Marks]
 - Three private variables representing *income*, *taxno* and *tax*.
 - A *constructor* which initializes *income* and *taxno*.
 - A public method that computes amount of tax for *income* and assign it to *tax*. *Income* up to Rs. 50000 is tax free. The tax value will be increased by 5% for every additional Rs. 50000.
(For example, *income* more than Rs. 50000 but less than Rs. 100000 is subject to 5% tax, *income* more than Rs. 100000 but less than Rs. 150000 is subject to 10% tax and more than Rs. 150000 is subject to 15% tax).
 - A default method that prints the attribute values and the calculated tax.