1. (a) (i) Explain the term encapsulation as used in OOP. (2 marks)

(ii) Distinguish between a class and a structure as used in OOP. (4 marks)

(b) Outline **four** rules of naming variables. (4 marks)

(c) A module II student intends to design a *class* for a bank account. Describe **two** operations that the student should incorporate in the class justifying your answer. (4 marks)

(d) Write a C++ program that implements a class named *modular* with the features outlined in table 1. (6 marks)

Feature	Description
State	Defined by two integers; a and b
Behaviour	Defined by an operation used to accept the values of a and b, determine and display the modulus of the two values.

Table 1

2. (a) Distinguish between *methods* and *messages* as used in OOP. (4 marks)

(b) (i) Define the following terms as used in OOP:

I. super class; (2 marks)

II. template class. (2 marks)

(ii) Explain the function of each of the following in OOP:

I. Object; (2 marks)

II. Scope resolution operator; (2 marks)

III. State transition diagrams. (2 marks)

(c) Figure 1 shows the cross-section of a water pipe used by engineers. Write a C++ program that would implement a class in determining the cross-sectional area of the pipe. The class should contain a constructor for initializing R and r as 10 and 7 respectively and an operation for determining the area. (6 marks)

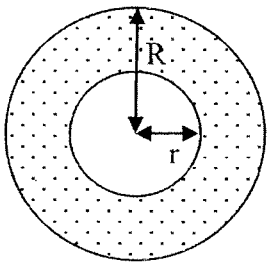

Figure 1

3. (a) A tutor would like to introduce OOP to module II students. Outline the **three** typical steps of programming in object oriented languages he would highlight in the introduction. (3 marks)

(b) (i) Differentiate between *static* and *dynamic* binding as applied in OOP. (4 marks)

(ii) Explain the fact that object orientation supports top-down programming. (2 marks)

(c) Justify the following statement. (4 marks)
"Data security is guaranteed in OOP"

(d) Interpret the following C++ program. (7 marks)

```
#include < iostream >
using namespace std;
int abs (int n);
long abs (long n);
double abs (double n);
int main( ) {
cout<< "Abs value of -10: "<< abs(-10)<< "\n";
cout<< "Abs value of -10L: "<< abs(-10L)<< "\n";
cout<<"Abs value of -10.01:"<<abs(-10.01)<<"\n";
return 0;
}
int abs (int n) {
cout << "In integer abs( )\n";
return (n<0)? -n: n;
}
long abs (long n) {
cout << "In long abs( )\n";
return (n<0)? –n:n; }
double abs (double n) {
cout << "In double abs( )\n";
return (n<0)? -n: n;
}
```

4. (a) (i) State **two** types of relationships common among classes in OOP. (2 marks)

(ii) Assuming C++ programming, outline the general format of a copy constructor. (2 marks)

(b) (i) With the aid of an illustration, describe *inline functions* as applied in OOP stating **one** advantage of their use. (4 marks)

(ii) Distinguish between *converter* and *modifier* operations as used in OOP. (4 marks)

(c) Write a C++ that implements a class with the dimensions of a trapezium as data members and a friend function used to read the values of the dimensions, determine and display the area of the trapezium. (8 marks)

Hint: Area of trapezium = ½ (a + b) h

5. (a) Describe **two** ways in which references can be used in OOP. (4 marks)

(b) (i) State **two** operators that can only be overloaded by a member function. (2 marks)

(ii) Explain **two** important restrictions that should be considered when overloading operators in OOP. (4 marks)

(c) With the aid of a C++ program segment, explain how you would overload a binary operator in a friend function. (4 marks)

(d) Write a C++ program that will create a class named rectangle with length and breadth as its member data. The class should also contain a constructor for initializing the dimensions and two member functions for determining the area and perimeter of an object with length and breadth as 10.5 and 7.0 respectively. (6 marks)

6. (a) (i) Explain the function of each of the following types of constructors:

I. default; (2 marks)

II. dynamic (2 marks)

(ii) Outline **four** characteristics of a destructor as used in OOP. (4 marks)

(b) Differentiate between *ofstream* and *ifstream* as used in files. (4 marks)

(c) (i) Inheritance is both an extension and a contraction. Justify this statement. (4 marks)

(ii) With the aid of an illustration, describe multilevel inheritance as used in OOP. (4 marks)

7. (a) (i) Outline **two** benefits of using inheritance in OOP. (2 marks)

(ii) Explain the importance of using protected members in parent classes. (2 marks)

(b) (i) State **two** processes that can be carried out in an existing file. (2 marks)

(ii) During object oriented programming sessions, a tutor instructed the students to use only the file organization methods supported by the application. Describe **two** such file organization methods. (4 marks)

(c) Using a pure virtual function, write a C++ program that would implement the class hierarchy shown in figure 2. The program should initialize the dimensions of the shape as a= 14 and b = 8, and output the area of the respective shapes. (10 marks)

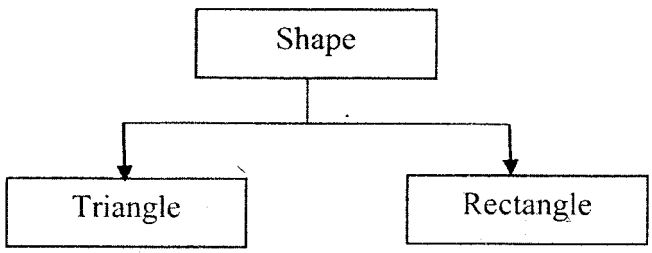

Figure 2

8. (a) (i) State **three** ways in which learning institution would cope with the emerging trends in OOP. (3 marks)

(ii) Distinguish between *universal* polymorphism and *ad-hoc* polymorphism as used in OOP. (4 marks)

(b) Joseph would like to apply inheritance in a project he is undertaking using OOP for a client. Explain **three** forms of inheritance he could use. (6 marks)

(c) Write a C++ program that would create an output file, write some string of text to the file, closes the file and then opens it again to read the information. The program should display the read information on the screen. (7 marks)

THIS IS THE LAST PRINTED PAGE.