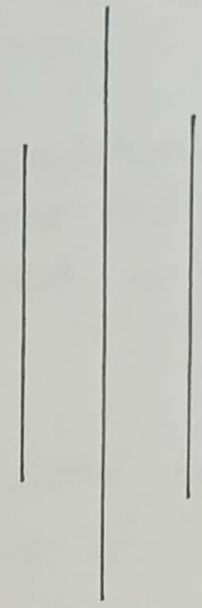
KATHMANDU UNIVERSITY

DHULIKHEL, KAVRE



Subject : COMP116

Assignment: 2

Submitted by:

Ashraya Kadel

CE 2022

Roll no: 25

Submitted to:

Rajani Chulyadyo

Department of Computer Science and Engineering

DATE OF SUBMISSION: 24/09/2023

Answer the following questions:

Q.17: Which drawbacks of structured programming are addressed by object-oriented programming?

Ans:

Structural programming language has some significant drawbacks over object oriented programming. Some of the drawbacks are as follows:

i) Emphasis on Data:

Structural programming paradigm gives importance to data operation whereas object oriented programming focuses on the data.

ii) Encapsulation

In structural programming, data is exposed to the whole program whereas object oriented programming fouses on providing relevant information and ensures data security.

(iii) Real-world Problems

Structure programming makes it difficult to tackle real world ~~programs~~ problems. object-oriented programming makes relating with real-world problems easier.

(iv) Large program:

Structural programming paradigm makes it difficult to manage ~~real world~~ large problems. We can code larger problems easily with object oriented programming.

〈Q.2〉: What is data encapsulation?

Ans:

Data encapsulation is the process of combining data members and member functions into a single unit.

It is one of the important feature of object-oriented programming.

In C++, data encapsulation is conducted with the use of classes and the combination of data and function into a unit is called object.

〈Q.3〉: What do you understand by access specifiers? Explain the different types of access specifiers in C++.

Ans:

Access specifiers are the modifiers that define how members of a class can be accessed.

C++ programming language has three different types of access specifiers.

There are three types of access specifiers. They are:

i) public
ii) private
iii) protected

i) public:

Public data members are accessible from outside the class. In UML diagram, the "+" sign represents public access specifier.

(ii) private:

Private access specifier prevents class members from beg being accessed from outside the class. It is symbolized by "-" sign in UML diagram.

(iii) protected:

Protected access specifiers make members inaccessible outside class but makes it possible to access the member through derived class/sub-class.

〈Q.4〉: Explain the this pointer.

Ans:

'this' pointer is a keyword that helps us initialize values to our member.

It stores the address of the class instance and helps us to ~~co~~ correctly access object members

Syntax: this -> class_member-name = value;

It is generally used with setters.

〈Q.5〉: Explain the different types of constructors.

Ans:

A class constructor is a special member function of a class that is executed whenever we create new objects of that class.

There are three types of constructors used in C++. They are as follows:

i) Default constructor
ii) Parameterized constructor
iii) Copy constructor

(i) Default constructor

Default constructor are the constructors that don't take any arguments and have no parameters. It is called when object is created and initializes the data members.

(ii) Parameterized constructor

Parameterized constructor are the constructors that take arguments and contain parameters. It is useful to set values to our data members.

(iii) Copy constructor

Copy constructor are special constructors that takes object as an argument and copies value of one object to another object.

<Q.67: Differentiate between constructor and destructor.

Constructor		Destructor
Constructor initializes objects.	-	Destructor cleanup and release resources
Syntax: class_name ( );	-	Syntax: ~class_name ( );
It is called when object is created.	-	It is called when object goes out of scope or explicitly deleted.
There can be multiple constructor with different parameters.	-	Typically there are one destructor per class.