Lesson 3: Object oriented programming languages

3.1. Introduction

In our previous lesson, we learnt about object oriented analysis and design. In this lesson, you will learn about object oriented programming languages. In this lesson, you will also learn about the structure of C++.

3.2. Lesson objectives

By the end of this lesson, you will be able to:

- Describe object oriented programming languages
- Describe the object oriented programming features
- Discuss the structure of C++

3.3. Lesson outline

This lesson is structured as follows:

3.1. Introduction
3.2. Lesson objectives
3.3. Lesson outline
3.4. OOPs
3.5. OOPs
3.6. OOP principles
3.7. Introduction to C++
3.8. Structure of C++
3.9. Revision questions
3.10. Summary
3.11. Suggested reading

3.5. Object oriented programming languages

These languages can be classified into two categories:

[a]. Object based programming languages: These languages support the following features: □ Data encapsulation

- Data hiding and access mechanisms
- Operator overloading
- Automatic initialization and clear-up of objects

Example: Ada

[b]. Object oriented programming languages: These languages supports all the features supported by object based programming languages as well as extra features such as inheritance and dynamic binding.

Examples: C++, small talk, object pascal, Java, c# etc.

3.6. Object oriented programming features

There are four basic features supported by object oriented language.

- Inheritance
- Abstraction(Information hiding)
- Polymorphism
- Encapsulation

[a]. Inheritance: This feature helps programmers to create new classes from existing classes. This results into code re-use.

[b]. Abstraction: This feature helps the programmer show necessary details only. For instance data can be made private, public or protected.

[c]. Polymorphism: This feature ensures that an object respond differently on different message.

[d]. Encapsulation: This feature ensures that class members such as data members and member functions are bound into units called objects.

3.7. Introduction to C++

C++ is an object oriented programming language. This means it can support the four basic principles of OOP. Like any programming language C++ language has three basic features:

[a]. Syntax: Rules that govern grammar.

[b]. Semantics: Rules that govern the meanings of the language statements.

[c]. Pragmatics: Rules that dictate how a language is used.

3.8. Structure of C++

A c++ program consists of the following components:

[a]. Header files: That usually consist of directives to the preprocessor. These statements begin with # symbol.
Examples:
#include<iostream>

[b]. Main function: All programs in C++ must have at least one function called main. This function must have a return type e.g. void or int.

[c]. Open and closing brackets: The are used to enclose a function block.

[d]. Variable declaration: Usually, the variables are declared at the top of the program. The variables can be local, global or formal parameters. We will discuss variables in our next lesson.

[e]. Cin: This is an object of istream class. It is usually followed by stream extraction operator '>>'. The variable to be extracted is placed to the right of the extraction operator.

[f]. Cout: This is an oject of the ostream class. It is usually followed by stream insertion operator '<<'. It outputs or prints the value or variable to the right of the insertion operator.

[g]. Return statement: Usually appears at the end of the main function but just before the closing bracket. It returns a value to the operating system when the program executes to completion.

[h]. Comments: A comment is an explanation of the code statement(s).

[i]. Using namespace std: This statement specifies that the namespace to be used is standard library.

```
#include<iostream>
using namespace std;
int main()
```

```
{                                          //execution begins
here     int x, y, sum;                          //declaration of
x,y,sum         cout << "Enter value of x:" << endl;
//statement to prompt user for x    cin >> x;
//statement to read x value  cout << "Enter value of y:" <<
endl;    //Statement to prompt user for y value   cin >> y;
//statement to read y value  sum = x + y;
//statement to calculate sum
 cout << "The sum is :" << sum << endl; //statement to display
sum      return 0;                              //returns a value
to OS
```

3.9. Revision questions

[a]. Explain the four features of object oriented languages

[b]. Describe any two advantages and two disadvantages of object oriented programming languages.

[c]. Write a program in C++ that prompts the user to enter three integers then calculates and displays the product and sum.

[d]. Describe the structure of a c++ program.

[e]. Explain how you develop a program in C++

3.10. Summary

In this lesson, you have learnt about object oriented programming languages- features, their advantages and disadvantages. Finally, you have learnt about the structure of a c++ program.

3.11. Suggested reading

[5]. Object oriented programming with C++ by E Balagurusamy 3rd ed; publisher: Tata Mcraw Hill

[6]. Sams teach yourself c++ in 24 hours by Jesse Liberty and Rogers Cadenhead. [7]. Object oriented programming in c++ by Joyce Farrel [8]. Object-oriented programming with c++ by Sourav Sahay.