PROGRAMMING CONCEPTS AND EMBEDDED PROGRAMMING IN C, C++ and JAVA:

Lesson-10: Object Oriented Language and C++

Object-oriented language features

- defining of the object or set of objects, which are common to similar objects within a program and between the many programs,
- defining the methods that manipulate the objects without modifying their definitions, and
- creation of multiple instances of the defined object or set of objects or new objects

Object-oriented language

- inheritance
- overloading of functions
- overriding of functions
- data encapsulation, and
- design of reusable components

Object Characteristics

1. An i*dentity* (a reference to a memory block that holds its state and behavior).

2. A *state* (its data, property, fields and attributes).

3. A *behavior* (method or methods that can manipulate the *state* of the object).

Procedure oriented language

Procedure oriented language —A large program in ‘C’ splits into the simpler functional blocks and statements. ‘C’ is called procedure oriented language.

Object Oriented Language Characteristics

- A large program in objected oriented language C++ or Java, splits into the logical groups (also known as ***classes***).
- Each class defines the data and functions (methods) of using data.
- Each class can inherit another class elements.

Object Oriented Language Characteristics

- A set of these groups (classes) then gives an application program of the Embedded System
- Each group has internal user-level fields for data and has methods of processing that data at these fields
- Each group can then create many objects by copying the group and making it functional

Object Oriented Language Characteristics

- Each object is functional. Each object can interact with other objects to process the user's data.
- The language provides for formation of classes by the definition of a group of objects having similar attributes and common behavior. A class *creates the **objects***. **An object is an instance of a class.**

Embedded Programming in C++

- C++ is an object oriented Program (OOP) language, which in addition, supports the procedure oriented codes of C.

- Program coding in C++ codes provides the advantage of objected oriented programming as well as the advantage of C and in-line assembly.

C++

- *struct* that binds all the member functions together in C. But a C++ *class* has object features. It can be extended and child classes can be derived from it. A number of child classes can be derived from a common class. This feature is called polymorphism. A class can be declared as public or private. The data and methods access is restricted when a class is declared private. *Struct* does not have these features.

C++

- A class binds all the member functions together for creating objects. The objects will have memory allocation as well as default assignments to its variables that are not declared *static*
- A class can derive (inherit) from another class also. Creating a *child* class from RTCSWT as a *parent* class creates a new application of the RTCSWT.
- Methods (C functions) can have same name in the inherited class. This is called *method overloading*

C++

- Methods can have the same name as well as the same number and type of arguments in the inherited class. This is called *method overriding*. These are the two significant features that are extremely useful in a large program.

C++

- Operators in C++ can be overloaded like in method overloading.
- For example, operators ++ and ! are overloaded to perform a set of operations.

Some disadvantages

- Lengthier Code when using Template, Multiple Inheritance (Deriving a class from many parents), Exceptional handling, Virtual base classes and classes for IO Streams

Ways to overcome the disadvantages

1) Declare private as many classes as possible. It helps in optimising the generated codes.
2) Use *char*, *int* and *boolean* (scalar data types) in place of the objects (reference data types) as arguments and use local variables as much as feasible.

Ways to overcome the disadvantages

3) Recover memory already used once by changing the reference to an object to NULL.

4) A ***special compiler for an embedded system*** can facilitate the disabling of specific features provided in C++.

Embedded C++ is a version of C++ that provides for a selective disabling of the above features

Ways to overcome the disadvantages

5) Use Embedded C++: It provides for less runtime overhead and less runtime library. The solutions for the library functions are available and ported in C directly.

6) The IO stream library functions in an embedded C++ compiler are also reentrant.

Ways to overcome the disadvantages

7) Using embedded C++ compilers or the special compilers make the C++ more powerful coding language than C for embedded systems due to the OOP features of software re-usability, extendibility, polymorphism, function overriding and overloading along portability of C codes and in-line assembly codes

Summary

We learnt

- C++ provides all the advantages of 'C' and of object oriented programming and is suitable for embedded systems
- Declare private as many classes as possible. It helps in optimising the generated codes.

We learnt

- Use *char*, *int* and *boolean* (scalar data types) in place of objects (reference data types) as arguments and use local variables as much as feasible.

We learnt

- Recover memory once already used by changing reference to an object to NULL, modularity, robustness, portability and platform independence.

We learnt

- Selectively remove the features of template, run time type identification, multiple Inheritances, exceptional handling, virtual base classes, IO streams and foundation classes.

End of Lesson 10 of Chapter 5