 **Introduction to C++ Programming**

The prime purpose of C++ programming was to add object orientation to the C programming language, which is in itself one of the most powerful programming languages. The core of the pure object-oriented programming is to create an object, in code, that has certain properties and methods. While designing C++ modules, we try to see whole world in the form of objects. For example a car is an object which has certain properties such as color, number of doors, and the like. It also has certain methods such as accelerate, brake, and so on.

Properties of C++ Programming:

* Data as well as operations are encapsulated in objects
* Information hiding is used to protect internal properties of an object
* Objects interact by means of message passing.
* In most object-oriented languages objects are grouped in classes
* Objects in classes are similar enough to allow programming of the classes, as opposed to programming of the individual objects
* Classes represent concepts whereas objects represent phenomena
* Classes are organized in inheritance hierarchies.Provides for class extension or specialization

**See the Video on Applications of C++ at following link:**

<https://www.softwaretestinghelp.com/cpp-applications/>

**Features of object-oriented programming**

**Object**

Object is a instance of class.

**Class**

Set of data & code of an object ->user-defined data type –>class.

Once a class has been defined, we can create any number of objects belonging to that class.

A class is collection of objects of similar type.

Syntax used to create an object is no different than the syntax used to create an integer object in C.

Example-- If fruit has been defined as a class, then the statement

fruit mango;

will create an object mango belonging to the class fruit.

**Data abstraction and Encapsulation**

Wrapping up of data & functions into a single unit (called class) -encapsulation.

Data is not accessible to the outside world

Functions which are wrapped in the class can access it.

Abstraction - act of representing essential features without background details.

**Information hiding**

Insulation of data from direct access by the program – data or information hiding.

Attributes are - data members because they hold information.

Functions are - member functions.

**Inheritance**

Process by which objects of one class acquire the properties of objects of another class.

Each derived class shares common characteristics with the class from which it is derived(fig.4).

Inheritance provides re-usability and expandability.

**Polymorphism**

Ability to take more than one form.

An operation may exhibit different behaviors in different instances.

Using a single function name to perform different types of tasks is known as function overloading(fig.5).

**Difference between object oriented and procedure-oriented programming**

**Summary**

In this lecture we have discussed about C++ programming language. We have discussed about some features of C++. Moreover, we have learnt about difference between Procedure-Oriented and Object-Oriented Programming

**FAQs**

Q1 What are the various features of Object-oriented programming?

Answer: Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

Q2 What are applications of C++ language?

Answer:

1) Games

C++ is close to the hardware, can easily manipulate resources, provide procedural programming over CPU intensive functions and is fast. It is also able to override the complexities of 3D games and provides multi-layer networking. All these benefits of C++ make it a primary choice to develop the gaming systems as well as game development suites.

2) GUI Based Applications

C++ can be used to develop most of the GUI based and desktop applications easily as it has got the required features.

3) Database Software

C++ is also used in writing database management software. The two most popular databases MySQL and Postgres are written in C++.

4) Operating Systems

The fact that C++ is a strongly typed and fast programming language makes it an ideal candidate for writing operating systems. In addition to this, C++ has a wide collection of system-level functions that also help in writing low-level programs.

5) Browsers

Browsers are mostly used in C++ for rendering purposes. Rendering engines need to be faster in execution as most people do not like to wait for the web page to be loaded. With the fast performance of C++, most browsers have their rendering software written in C++.

6) Advanced Computation And Graphics

C++ is useful in developing an application that requires high-performance image processing, real-time physical simulations, and mobile sensor applications that need high performance and speed.

7) Banking Applications

As C++ aids in concurrency, it becomes the default choice for banking applications that require multi-threading, concurrency, and high performance.

8) Cloud/Distributed System

Cloud storage systems that are extensively used nowadays work close to the hardware. C++ becomes a default choice for implementing such systems as it is close to the hardware. C++ also provides multithreading support that can build concurrent applications and load tolerance.

9) Compilers

Compilers of various high-level programming languages are written either in C or C++. The reason is that both C and C++ are low-level languages that are close to hardware and are able to program and manipulate the underlying hardware resources.

10) Embedded Systems

Various embedded systems like smartwatches, medical equipment systems use C++ to program as it is closer to the hardware level and can provide a lot of low-level function calls when compared to the other high-level programming languages.

11) Enterprise Software

C++ is used in developing many enterprise software as well as advanced applications like flight simulation and radar processing.

12) Libraries

When we require very high-level mathematical computations, performance and speed become important. Hence most of the libraries use C++ as their core programming language. Most high-level machine language libraries use C++ as backend.

**References:**

**Books:**

* EBalagurusamy., “Object Oriented Programming in C++”, Tata McGraw-Hill.
* RobertLafore, “Object Oriented Programming in C++”, Waite Group.

**Reference Books:**

* Herbert Schildt , “C++- The Complete Reference”,Tata McGraw-Hill 2003, New Delhi.
* Bjarne Stroustrup: “The C++ Programming Language”(4th Edition). Addison-Wesley.
* Ravichandran , “Programming with C++”,TataMcGraw-Hill Education.
* Joyce M. Farrell,” Object Oriented Programming UsingC++”, Learning.
* Programming Languages: Design and Implementation(4th Edition), by Terrence W. Pratt, Marvin V. Zelkowitz, Pearson.
* Programming Language Pragmatics, Third Edition, byMichael L. Scott, Morgan Kaufmann.

**Websites:**

* https://www.sanfoundry.com/cplusplus-programming-questions-answers-basics/
* https://www.cppbuzz.com/100-mcq-on-c++
* https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/

**YouTube Links:**

* https://www.softwaretestinghelp.com/cpp-applications/