Welcome to ineuron.ai

C++ Job Preparation

Description:

C++ Interview-ready course has been created specifically to familiarize you with the types of questions you may encounter during the interview. This course is for the people who want to prepare for their interview after learning all the initial concepts & theories with Project Building. After completing this course, you will be confident enough to crack any discussion related to C++.

Start Date:

Doubt Clear Time:

Course Time:

Features:

# Challenges

# Downloadable resources

# Quizzes

# Assignments in each module

# Completion certificate

What we learn:

# Tackle difficult interview questions

# Present projects in interview

# Create application

# Distribution

# Packaging

# Architecture design

# Bug testing

# Modular coding

# Object-oriented programming STL (Structured Template Library) for competitive progr

Requirements:

# Prior knowledge of C++ language

# A system with internet connection.

# Dedication

Instructor:

Name:

Umang Pincha

Description:

Data Warehouse/Business Intelligence Developer @ Amdocs & a Competitive Programmer. He is also having a good knowledge in C , C++ , Data Structure , Python , ML I can easily make websites on it and have done PG in Machine

Learning and Artificial Learning from NIT, WARANGAL

>Essentials and Fundamentals:

>>What is the difference between new/delete and malloc/free?

>>What is the difference between new and malloc?

>>What is the difference between delete and delete[]?

>>What is difference between malloc()/free() and new/delete?

>>What is the difference between ""new"" and ""operator new""?

>>What is Memory alignment?

>>Is there a way to force new to allocate memory from a specific memory area?

>>How does free know the size of memory to be deleted.?

>>int *i = (int *)malloc(12); followed by free(i); how did free function call know how much

>>How do I allocate multidimensional arrays using new ones?

>>Can I free() pointers allocated with new? Can I delete pointers allocated with malloc()

>>Why should I use new instead of trustworthy old malloc()?

>>Can I use realloc() on pointers allocated via new?

>>Do I need to check for NULL after p = new Fred()?

>>How can I convince my (older) compiler to automatically check new to see if it returns

>>Do I need to check for NULL before deleting p?

>>What are the two steps that happen when I say delete p?

>>In p = new Fred(), does the Fred memory ""leak"" if the Fred constructor throws an ex

>>How do I allocate/deallocate an array of things?

>>What if I forget the [] when deleting the array allocated via new T[n]?

>>Can I drop the [] when deleting an array of some built-in type (char, int, etc.)?

>>After p = new Fred[n], how does the compiler know there are n objects to be destructo

>>Is it legal (and moral) for a member function to say delete this?

>>C++ Basics & Conditional Statements

>>How do you link a C++ program to C functions?

>>Is there anything you can do in C++ that you cannot do in C?

>>What are the differences between a struct in C and in C++?

>>What does extern ""C"" int func(int *, Foo) accomplish?

>>What are the access privileges in C++? What is the default access level?

>>How does C++ help with the tradeoff of safety vs usability?

>>What are the benefits of operator overloading?

>>What are some examples of operator overloading?

>>But operator overloading makes my class look ugly; isn't it supposed to make my cod

>>Can I overload operator==, so it lets me compare two char[] using a string compariso

>>Can I create an operator** for ""to-the-power-of"" operations?

>>Okay, that tells me the operators I can override; which operators should I override?

>>What are some guidelines / ""rules of thumb"" for overloading operators?

>>Base Class Pointer & Derived Class Object

>>What is a dangling pointer?

>>What is Memory Leak?

>>What is an auto pointer?

>>What issue do auto_ptr objects address?

>>What is a smart pointer?

>>Is there any problem with the following : char*a=NULL; char& p = *a;?

>>What is the difference between a pointer and a reference?

>>What is the difference between const char *myPointer and char *const myPointer?

>>When should I use references, and when should I use pointers?

>String:

>>How to convert an integer to a string?

>>Programs related to string.

>OOPS:

>>Is it possible to have a Virtual Constructor? If yes, how? If not, why not possible?Is it

>Polymorphism & Virtual Function:

>>What is Polymorphism?

>>What is the problem with Runtime type identification?

>>What is virtual function?

>>What is a "pure virtual" member function?

>>How are virtual functions implemented in C++?

>>What is pure virtual function? or what is an abstract class?

>>How Virtual functions call up is maintained?

>>What is a virtual destructor?

>Inheritance:

>>What is inheritance?

>>When should you use multiple inheritance?

>>Explain the ISA and HASA class relationships. How would you implement each in a c

>>When is a template a better solution than a base class?

>>What is multiple inheritance (virtual inheritance)? What are its advantages and disadv

>>What a derived class inherits or doesn't inherit?