OBJECT-ORIENTED PROGRAMMING WITH C++ PDF, EPUB, EBOOK

Sourav Sahay | 486 pages | 14 Dec 2006 | OUP India | 9780195681529 | English | New Delhi, India

Learn Object Oriented Programming (OOP) in C++ | Full Video Course

Faster than videos. No set-up required. Progress you can show. Hands-on coding environments. Course Contents Hide All. Hide All Lessons. Assessment 1. Data Hiding. Composition, Aggregation and Association. Assessment 2. Join , learners from companies like. Teams of every size choose Educative for Business. When a class is defined, no memory is allocated but when it is instantiated i.

When a program is executed the objects interact by sending messages to one another. Each object contains data and code to manipulate the data. Encapsulation : In normal terms, Encapsulation is defined as wrapping up of data and information under a single unit. In Object-Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulate them. Consider a real-life example of encapsulation, in a company, there are different sections like the accounts section, finance section, sales section etc. The finance section handles all the financial transactions and keeps records of all the data related to finance. Similarly, the sales section handles all the sales-related activities and keeps records of all the sales. Now there may arise a situation when for some reason an official from the finance section needs all the data about sales in a particular month. In this case, he is not allowed to directly access the data of the sales section. He will first have to contact some other officer in the sales section and then request him to give the particular data.

This is what encapsulation is. Skip to content. A class is a description of the common characteristics shared by all objects of the same type. It is in fact a user-defined data type, with all the rights and privileges from a language point of view given to standard data types such as int, float, or double. All rights reserved. KIP R. Share Embed e-mail. Private Account Public.

Object Oriented Programming in C++ - GeeksforGeeks

Procedural programming is about writing procedures or functions that perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions. You should extract out the codes that are common for the application, and place them at a single place and reuse them instead of repeating it. When the individual objects are created, they inherit all the variables and functions from the class. You will learn much more about classes and objects in the next chapter. We just launched W3Schools videos. Get certified by completing a course today! If you want to report an error, or if you want to make a suggestion, do not hesitate to send us an e-mail:.

Arithmetic Assignment Comparison Logical. Boolean Values Boolean Expressions. Create References Memory Address. In the above example of class Car, the data member will be speed limit, mileage etc and member functions can apply brakes, increase speed etc. Object: An Object is an identifiable entity with some characteristics and behaviour. An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated i. When a program is executed the objects interact by sending messages to one another.

Each object contains data and code to manipulate the data. Encapsulation : In normal terms, Encapsulation is defined as wrapping up of data and information under a single unit. In Object-Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulate them. Consider a real-life example of encapsulation, in a company, there are different sections like the accounts section, finance section, sales section etc. The finance section handles all the financial transactions and keeps records of all the data related to finance. Similarly, the sales section handles all the sales-related activities and keeps records of all the sales. Now there may arise a situation when for some reason an official from the finance section needs all the data about sales in a particular month. In this case, he is not allowed to directly access the data of the sales section. He will first have to contact some other officer in the sales section and then request him to give the particular data.

This is what encapsulation is. Skip to content. Change Language. Related Articles. Object Oriented Programming. Exception Handling.

Learn Object-Oriented Programming in C++ - Learn Interactively

Online Marketing. Verbal Ability. Interview Questions. Company Questions. Artificial Intelligence. Cloud Computing. Data Science. Angular 7. Machine Learning. Data Structures. Operating System. Computer Network. Compiler Design. Computer Organization. Discrete Mathematics. Ethical Hacking. Computer Graphics. Software Engineering. Web Technology. Cyber Security. C Programming. Control System. Data Mining. Data Warehouse. Javatpoint Services JavaTpoint offers too many high quality services. It simplifies the software development and maintenance by providing some concepts: Object Class Inheritance Polymorphism Abstraction Encapsulation Object Any entity that has state and behavior is known as an object. Class Collection of objects is called class. It is a logical entity. Inheritance When one object acquires all the properties and behaviours of parent object i. Polymorphism When one task is performed by different ways i.

Now again, both Male and Female, performs some common functions, but there are some specifics to both, which is not valid for the other. For example : A Female can give birth, while a Male cannot, so this is only for the Female. Human Anatomy is interesting, isn't it? Here we will try to explain all the OOPS concepts through this example and later we will have the technical definitons for all this. Here we can take Human Being as a class.

A class is a blueprint for any functional entity which defines its properties and its functions. Like Human Being, having body parts, and performing various actions. Considering HumanBeing a class, which has properties like hands, legs, eyes etc, and functions like walk, talk, eat, see etc. Male and Female are also classes, but most of the properties and functions are included in HumanBeing , hence they can inherit everything from class HumanBeing using the concept of Inheritance. When we say, Human Being, Male or Female, we just mean a kind, you, your friend, me we are the forms of these classes.

We have a physical existence while a class is just a logical definition. We are the objects. Abstraction means, showcasing only the required things to the outside world while hiding the details. Continuing our example, Human Being's can talk, walk, hear, eat, but the details are hidden from the outside world. We can take our skin as the Abstraction factor in our case, hiding the inside mechanism. This concept is a little tricky to explain with our example. Our Legs are binded to help us walk. Our hands, help us hold things. This binding of the properties to functions is called Encapsulation. Polymorphism is a concept, which allows us to redefine the way something works, by either changing how it is done or by changing the parts using which it is done.

Both the ways have different terms for them. If we walk using our hands, and not legs, here we will change the parts used to perform something. Hence this is called Overloading. And if there is a defined way of walking, but I wish to walk differently, but using my legs, like everyone else. Then I can walk like I want, this will be called as Overriding. Objects are the basic unit of OOP. They are instances of class, which have data

members and uses various member functions to perform tasks. It is similar to structures in C language. Class can also be defined as user defined data type but it also contains functions in it. So, class is basically a blueprint for object. Abstraction refers to showing only the essential features of the application and hiding the details.

This can be done using access specifiers. It can also be said data binding. Encapsulation is all about binding the data variables and functions together in class. Inheritance is a way to reuse once written code again and again.

C++ Object Oriented

Get certified by completing a course today! If you want to report an error, or if you want to make a suggestion, do not hesitate to send us an e-mail:. Arithmetic Assignment Comparison Logical. Boolean Values Boolean Expressions. Create References Memory Address. Create Pointers Dereferencing Modify Pointers. See More. Coding is no different. Practice as you learn with live code environments inside your browser. Videos are holding you back. The average video tutorial is spoken at words per minute, while you can read at Built in assessments let you test your skills. Completion certificates let you show them off. All rights reserved. Search Courses. For Business. Get Unlimited Get Educative Unlimited.

Educative for Business. To use an interface, you cannot create objects. Instead, you need to implement that interface and define the methods for their implementation. Static Polymorphism is commonly known as the Compile time polymorphism. Static polymorphism is the feature by which an object is linked with the respective function or operator based on the values during the compile time. Static or Compile time Polymorphism can be achieved through Method overloading or operator overloading. Dynamic Polymorphism or Runtime polymorphism refers to the type of Polymorphism in OOPs, by which the actual implementation of the function is decided during the runtime or execution. The dynamic or runtime polymorphism can be achieved with the help of method overriding. Overloading is a compile-time polymorphism feature in which an entity has multiple implementations with the same name.

For example, Method overloading and Operator overloading. Whereas Overriding is a runtime polymorphism feature in which an entity has the same name, but its implementation changes during execution. For example, Method overriding. An abstract class is a special class containing abstract methods. The significance of abstract class is that the abstract methods inside it are not implemented and only declared. So as a result, when a subclass inherits the abstract class and needs to use its abstract methods, they need to define and implement them. Interface and abstract class both are special types of classes that contain only the methods declaration and not their implementation.

But the interface is entirely different from an abstract class. The main difference between the two is that, when an interface is implemented, the subclass must define all its methods and provide its implementation. Whereas when an abstract class is inherited, the subclass does not need to provide the definition of its abstract method, until and unless the subclass is using it. Access specifiers, as the name suggests, are a special type of keywords, which are used to control or specify the accessibility of entities like classes, methods, etc.

These access specifiers also play a very vital role in achieving Encapsulation - one of the major features of OOPs. An exception can be considered as a special event, which is raised during the execution of a program at runtime, that brings the execution to a halt. The reason for the exception is mainly due to a position in the program, where the user wants to do something for which the program is not specified, like undesirable input.

No one wants its software to fail or crash. Exceptions are the major reason for software failure. The exceptions can be handled in the program beforehand and prevent the execution from stopping. This is known as exception handling. So exception handling is the mechanism for identifying the undesirable states that the program can reach and specifying the desirable outcomes of such states. Try-catch is the most common method used for handling exceptions in the program. Object-oriented programming revolves around entities like objects. Each object consumes memory and there can be multiple objects of a class. So if these objects and their memories are not handled properly, then it might lead to certain memory-related errors and the system might fail.

Garbage collection refers to this mechanism of handling the memory in the program. Through garbage collection, the unwanted memory is freed up by removing the objects that are no longer needed. Java applications are based on Object-oriented programming models or OOPs concept, and hence they cannot be implemented without it. Reason: The above program demonstrates Multiple inheritances. Reason: Firstly the static block inside the main-method calling class will be implemented.

Then the main method is called, and now the sequence is kept as expected. Reason: ClassA contains a conversion constructor. Due to this, the objects of ClassA can have integer values. So the statement g 20 works. Also, ClassB has a conversion operator overloaded. So the statement g b also works. Reason: Here the main method is overloaded. But JVM only understands the main method which has a String[] argument in its definition. Hence Main1 is printed and the overloaded main method is ignored. Hence it results in wastage of space and a large size output. It can be reduced with the help of a virtual base class. Reason: The above program implements a Multi-level hierarchy.

So the program is linearly searched up until a matching function is found. Here, it is present in both classes A and B. Which language among the following supports classes, but does not support the concept of Polymorphism? Before you go! Download PDF. Enter the name of your college. Computer Science. Information Technology. Mathematics and Computing. Before After Enter company name. Forgot Password. What is meant by the term OOPs? What is the need for OOPs? What are some major Object Oriented Programming languages? What are some other programming paradigms other than OOPs? What is meant by Structured Programming? What are the main features of OOPs? What are some advantages of using OOPs? Why is OOPs so popular? What is a class? What is an object? What is encapsulation? What is Polymorphism? What is Compile time Polymorphism and how is it different from Runtime Polymorphism? What is meant by Inheritance?

What is Abstraction? How much memory does a class occupy? Is it always necessary to create objects from class? What is a constructor? What is a copy constructor? What is a destructor? Are class and structure the same? If not, what's the difference between a class and a structure? Explain Inheritance with an example?

Selected Takes : Film Editors on Editing
Tap Dancing on the Roof : Sijo (Poems)
Reputation Management : The Key to Successful Public Relations and Corporate Communication