Encapsulation is the process of enclosing all critical information inside an object and only revealing a subset of it to the outside world.

Abstraction refers to the user’s interaction with just a subset of an object’s characteristics and operations. To access a complicated item, abstraction uses simpler, high-level techniques.

Polymorphism refers to the creation of items that have similar behavior. For example, objects may override common parent behaviors with particular child behaviors through inheritance

Method overriding is used in runtime polymorphism. When a child class overrides a parent class’s method, the child class might offer an alternative implementation.

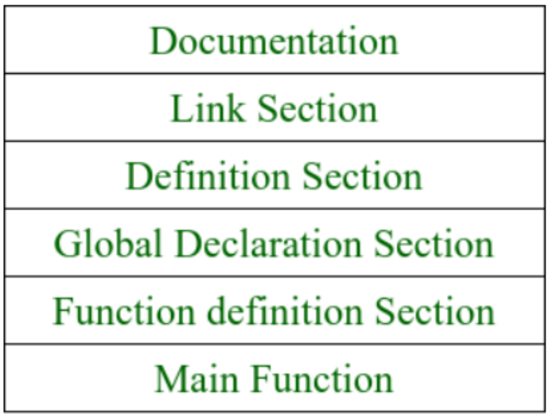
Method overloading is used in Compile Time Polymorphism. Although two methods or functions may have the same name, the number of arguments given into the method call may vary. Therefore, depending on the number of parameters entered, you may obtain different results.

An object is a self-contained segment with the attributes and processes needed to make data usable in programming terms.

In the oops concept, a class is a construct that is used to describe an individual type. The class is instantiated into instances of itself – referred to as class instances or simply objects. A class defines ingredient members that allow its instances to have position and behavior.

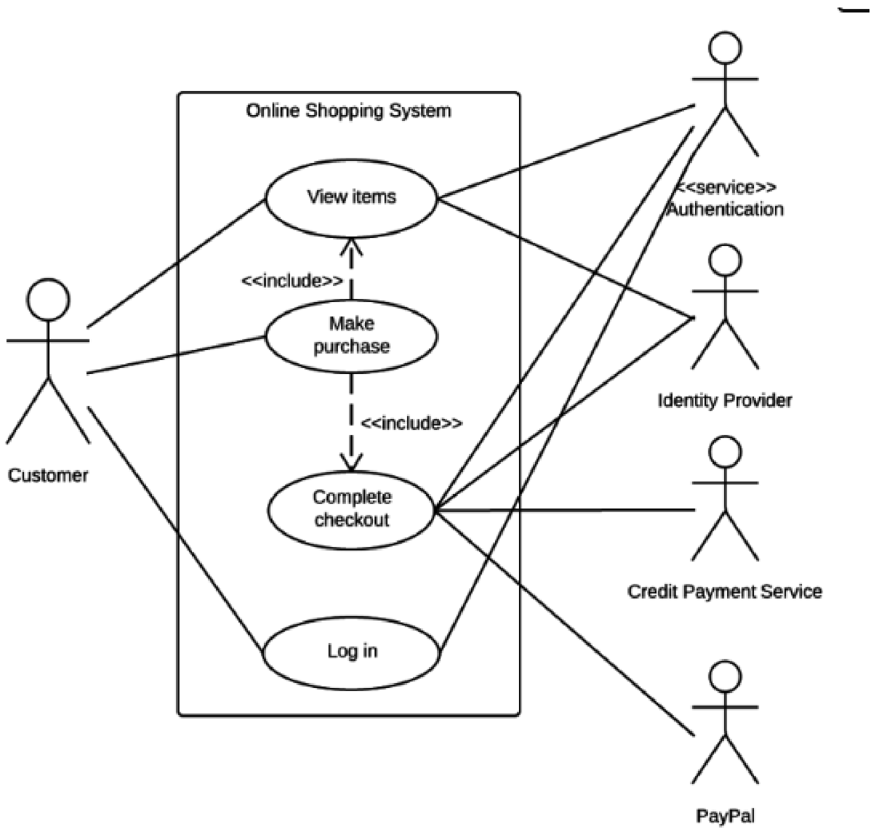
Constructors in most object-oriented languages have the same name as the class and are public. Constructors may be overloaded, which means that multiple argument lists can be used with the same name.

The C++ program is written using a specific template structure. The structure of the program written in C++ language is as follows:

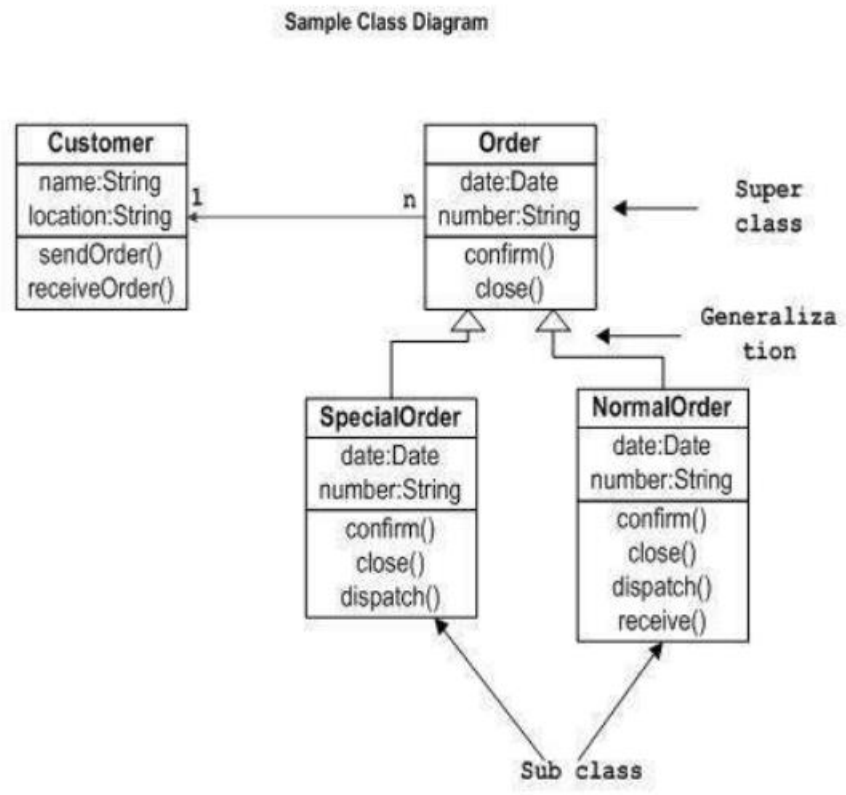


Documentation
Link Section
Definition Section
Global Declaration Section
Function definition Section
Main Function

USE CASE

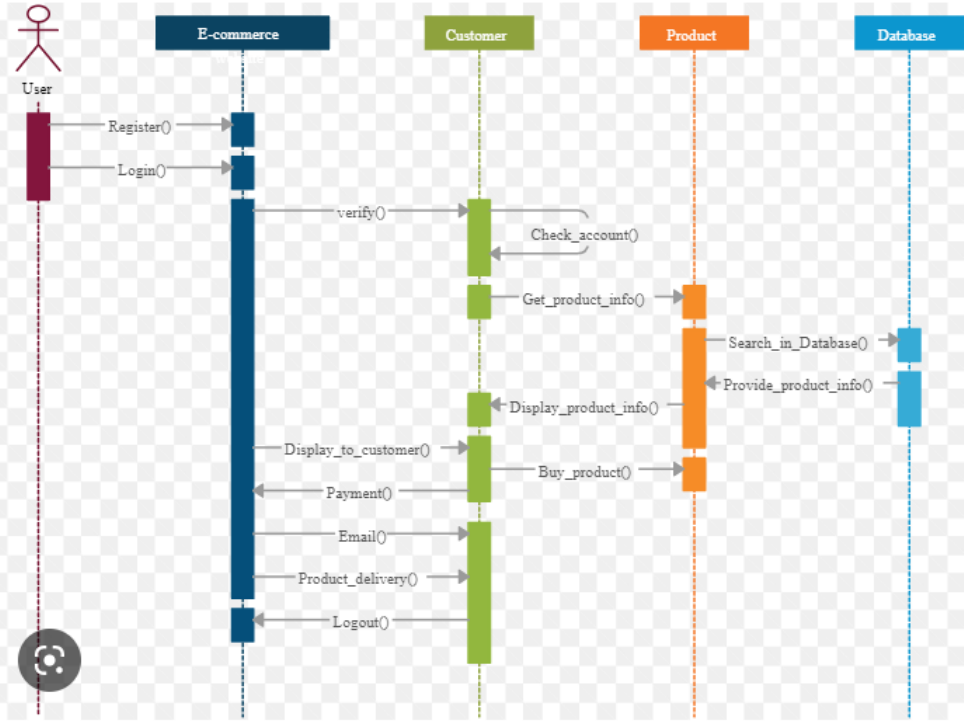


CLASS DIAGRAM

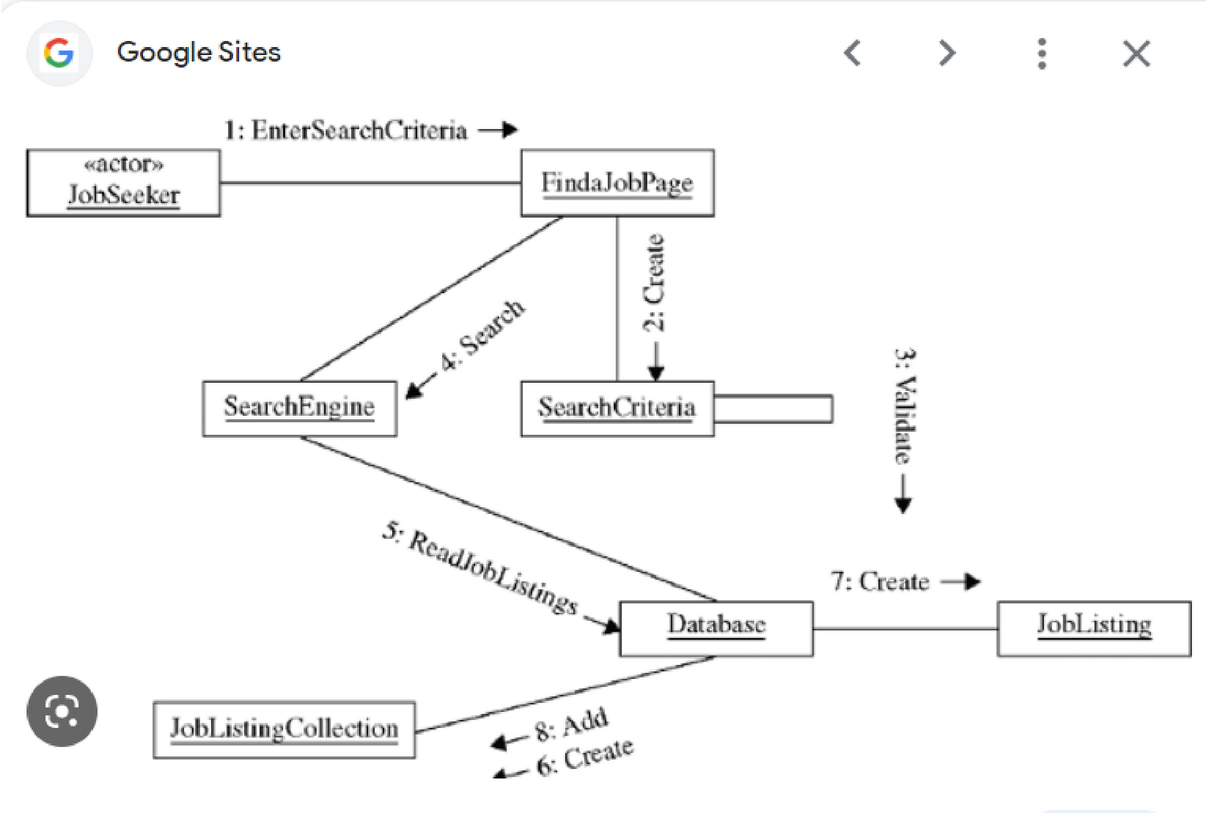


UNIT 2

SEQUENCE DIAGRAM



COLLABORATION DIAGRAM



Constructors

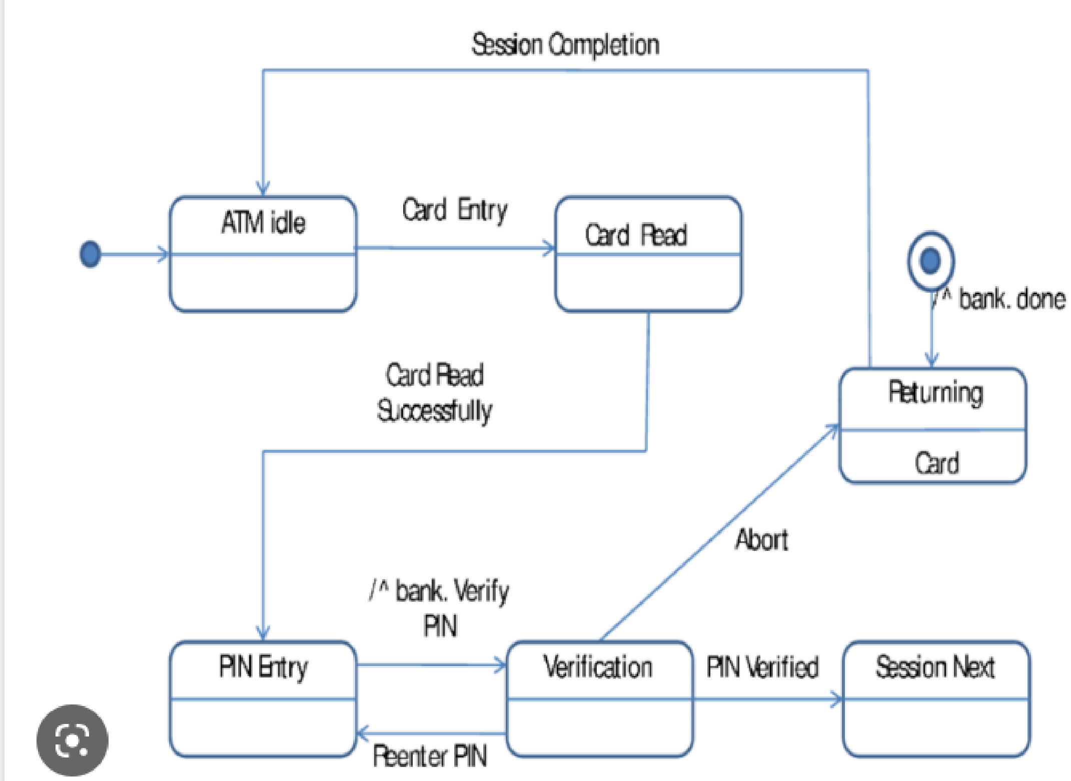
https://www.geeksforgeeks.org/constructors-c/

Constructor overloading

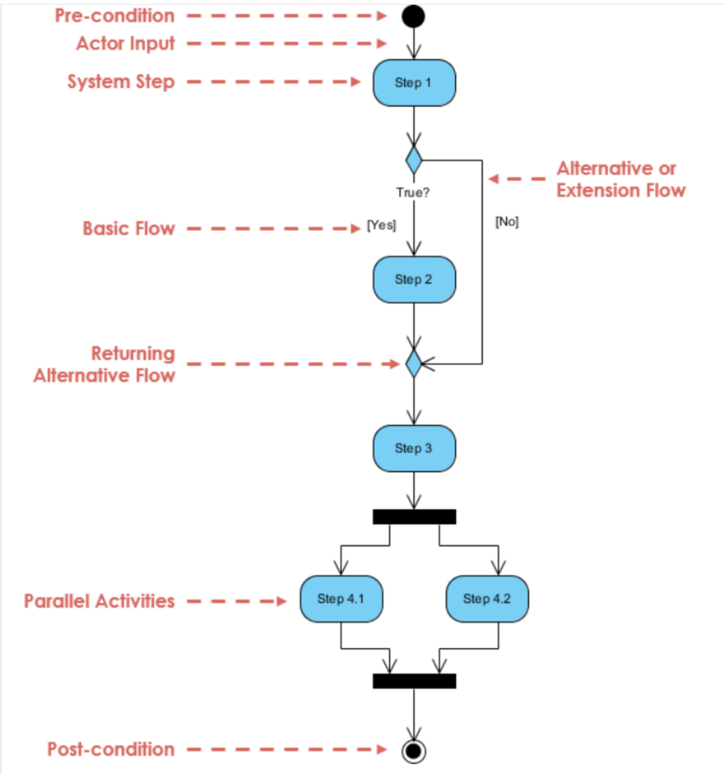
https://www.geeksforgeeks.org/constructor-overloading-c/

UNIT 3

State chart



Activity

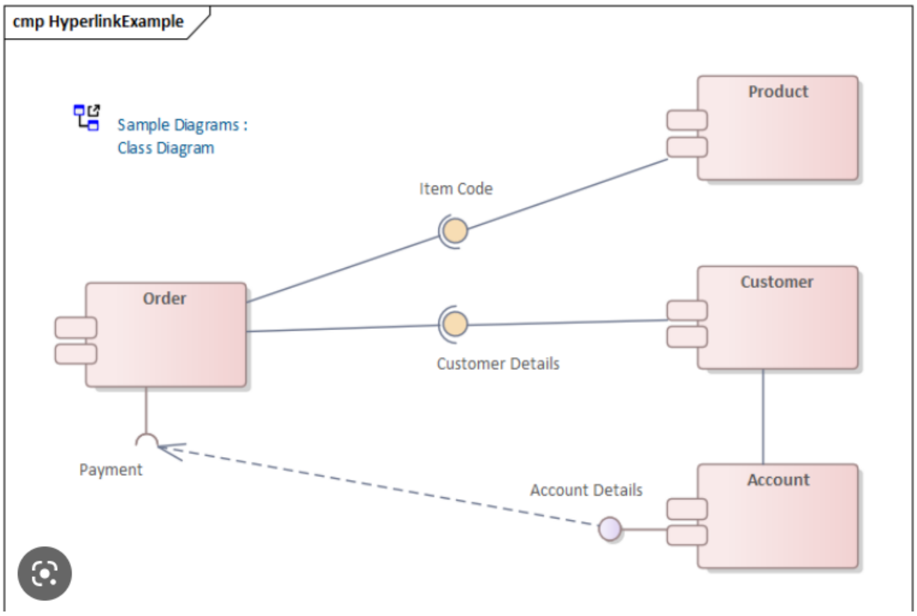


Inheritance with example

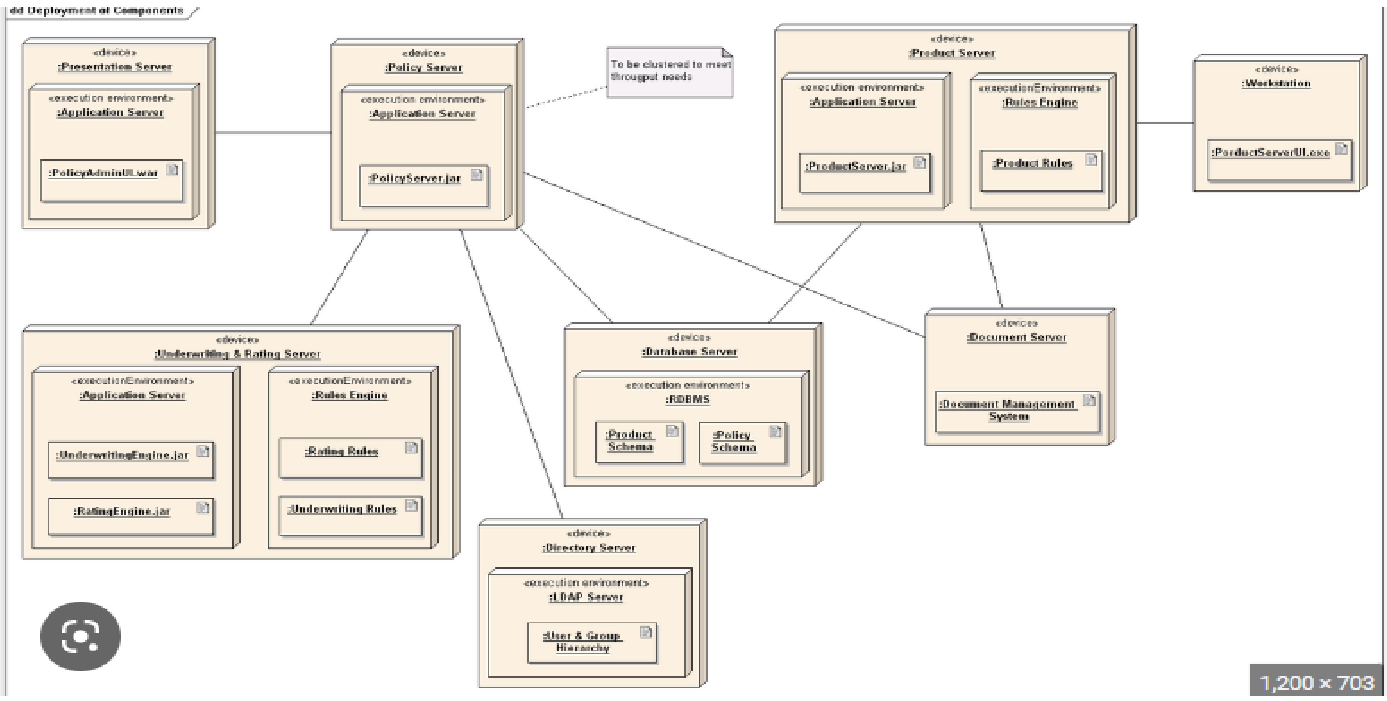
https://www.simplilearn.com/tutorials/cpp-tutorial/types-of-inheritance-in-cpp

UNIT 4

Component



Deployment


dd Deployment of Components
«device»
:Presentation Server
«execution environment»
:Application Server
:PolicyAdminUI.war
«device»
:Policy Server
«execution environment»
:Application Server
:PolicyServer.jar
To be clustered to meet througput needs
«device»
:Product Server
«execution environment»
:Application Server
:ProductServer.jar
«executionEnvironment»
:Rules Engine
:Product Rules
«device»
:Workstation
:PorductServerUI.exe
«device»
:Underwriting & Rating Server
«executionEnvironment»
:Application Server
:UnderwritingEngine.jar
:RatingEngine.jar
«executionEnvironment»
:Rules Engine
:Rating Rules
:Underwriting Rules
«device»
:Database Server
«execution environment»
:RDBMS
:Product Schema
:Policy Schema
«device»
:Document Server
:Document Management System
«device»
:Directory Server
«execution environment»
:LDAP Server
:User & Group Hierarchy
1,200 × 703