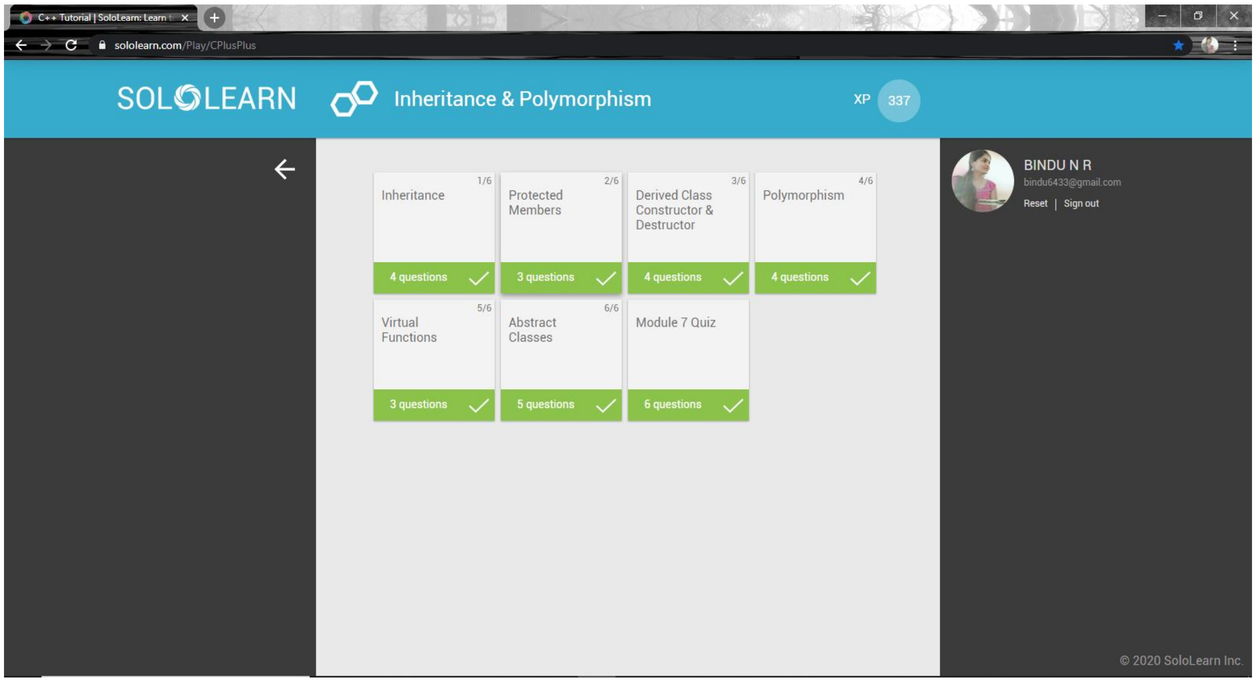
DAILY ASSESSMENT FORMAT

Course:	**C++**	**Name:**	**Bindu.N.R**
Link :	https://www.sololearn.com/	**USN:**	**4AL17EC101**
Org By:	**SOLOLEARN**	**Semester & Section:**	**6-B**
Github Repository:	**bindunr-python**	**Date:**	**27/06/2020**

Topic Completed Today

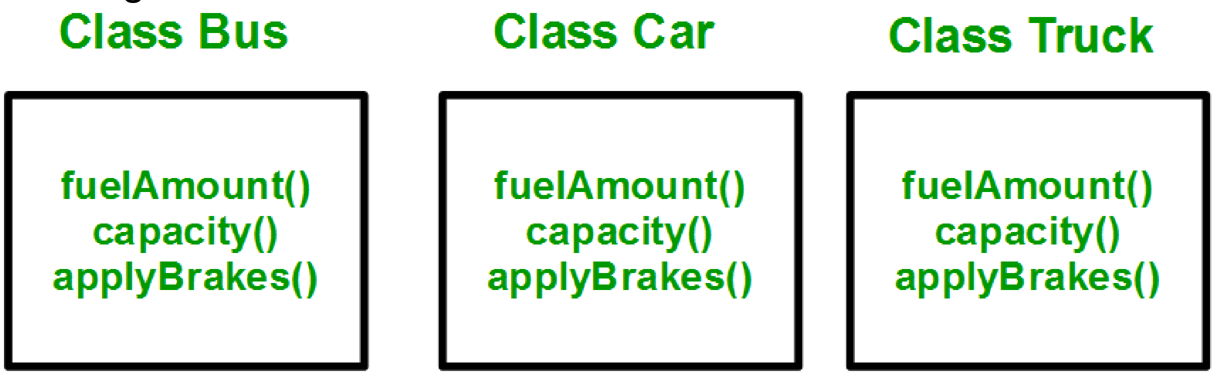

C++ Tutorial | SoloLearn: Learn
sololearn.com/Play/CPlusPlus
SOLOLEARN
Inheritance & Polymorphism
XP 337
Inheritance 1/6 4 questions
Protected Members 2/6 3 questions
Derived Class Constructor & Destructor 3/6 4 questions
Polymorphism 4/6 4 questions
Virtual Functions 5/6 3 questions
Abstract Classes 6/6 5 questions
Module 7 Quiz 6 questions
BINDU N R
bindu6433@gmail.com
Reset | Sign out
© 2020 SoloLearn Inc.

Inheritance in C++

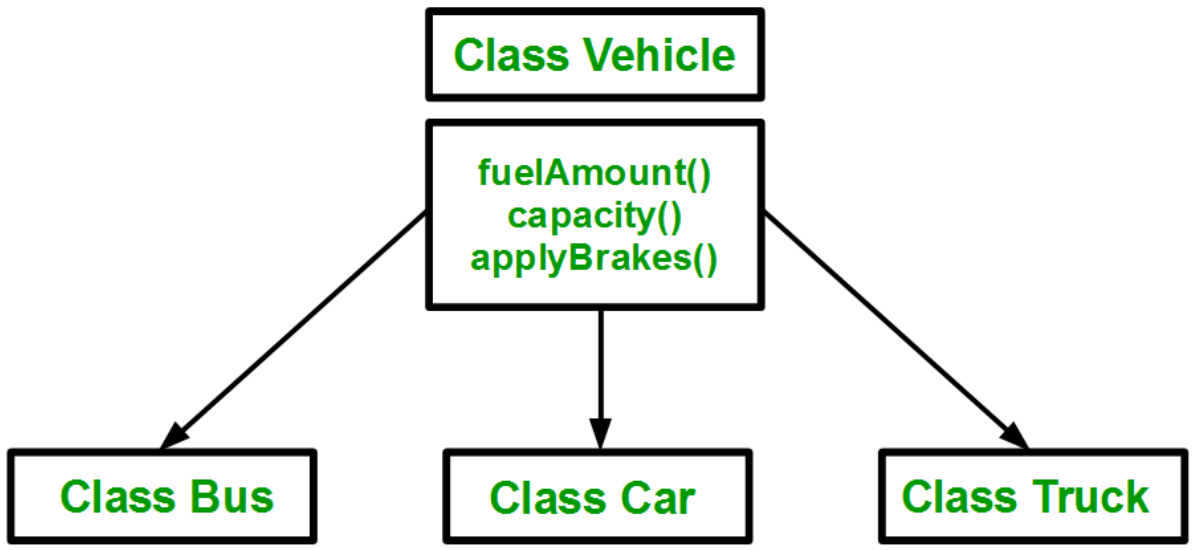
The capability of a class to derive properties and characteristics from another class is called **Inheritance**. Inheritance is one of the most important feature of Object Oriented Programming.

Sub Class: The class that inherits properties from another class is called Sub class or Derived Class.
Super Class: The class whose properties are inherited by sub class is called Base Class or Super class.

Consider a group of vehicles. You need to create classes for Bus, Car and Truck. The methods fuelAmount(), capacity(), applyBrakes() will be same for all of the three classes. If we create these classes avoiding inheritance then we have to write all of these functions in each of the three classes as shown in below figure:

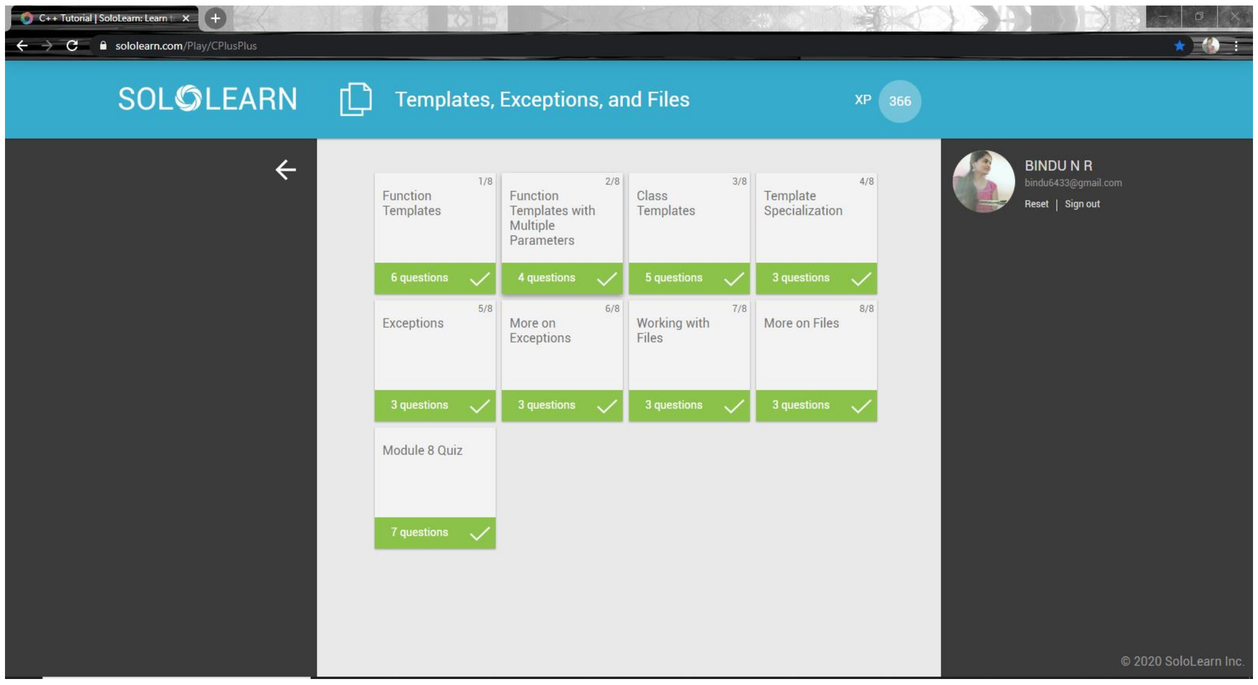


You can clearly see that above process results in duplication of same code 3 times. This increases the chances of error and data redundancy. To avoid this type of situation, inheritance is used. If we create a class Vehicle and write these three functions in it and inherit the rest of the classes from the vehicle class, then we can simply avoid the duplication of data and increase re-usability. Look at the below diagram in which the three classes are inherited from vehicle class:



Using inheritance, we have to write the functions only one time instead of three times as we have inherited rest of the three classes from base class(Vehicle).

Templates, Exceptions and Files


C++ Tutorial | SoloLearn: Learn t
sololearn.com/Play/CPlusPlus
SOLOLEARN

Templates, Exceptions, and Files

XP 366

BINDU N R
bindu6433@gmail.com
Reset | Sign out

1/8 Function Templates — 6 questions ✓
2/8 Function Templates with Multiple Parameters — 4 questions ✓
3/8 Class Templates — 5 questions ✓
4/8 Template Specialization — 3 questions ✓
5/8 Exceptions — 3 questions ✓
6/8 More on Exceptions — 3 questions ✓
7/8 Working with Files — 3 questions ✓
8/8 More on Files — 3 questions ✓
Module 8 Quiz — 7 questions ✓

© 2020 SoloLearn Inc.

Function Templates

Functions and classes help to make programs easier to write, safer, and more maintainable.

However, while functions and classes do have all of those advantages, in certain cases they can also be somewhat limited by C++'s requirement that you specify types for all of your parameters.

The function works as expected, but is limited **solely to integers**.

It becomes necessary to write a new function for each new type, such as doubles.

Function templates give us the ability **to write one version of sum() to work with parameters of any type**.

With function templates, the basic idea is to avoid the necessity of specifying an exact type for each variable. Instead, C++ provides us with the capability of defining functions using **placeholder** types, called **template type parameters**.