# Assignment 4

CSE 681 SOFTWARE MODELING & ANALYSIS

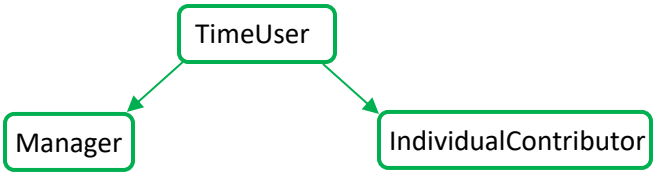
PROF. GREGORY WAGNER

9/13/2022

Anthony Redamonti
SYRACUSE UNIVERSITY

# Enterprise Computing Case Study: Timekeeping Software

If I were to create timekeeping enterprise software, I would implement a class structure so that managers and individual contributors are derived classes of a shared base class: TimeUser. The TimeUser base class would have private data member variables like Department, Business Location, First Name, Last Name, etc. which would all be set by the initialization constructor. It would also have a pure virtual function called printMemberData.




The initialization constructors of the derived classes would require the information needed to call the base class constructor. The IndividualContributor class would have private data member variables like TimeToday, TotalTimeThisPeriod, VacationTimeAccrued, and NetPay among others. It would have public "get" methods for these variables and a public "setTimeToday" method, which would set the TimeToday variable and add to the TotalTimeThisPeriod variable. It would also have the method "requestTimeOff," which would alter the private VacationTimeAccrued variable.

After the IC alters any data, it would have the ability to enter this data into a shared database (server) acting as a client. The IC class would implement the pure virtual printMemberData function by displaying the following information to the console: the first and last name, business location, department, and individual contributor badge number.

The Manager class (client) would have access to the shared server. After reading the entered data by the IC class, it would have the ability to reject, approve, or make alterations to the entries. The Manager class would implement the printMemberData function by printing the following information to the console: first and last name, business location, department, number of employees working for the manager, and the manager badge number.

Polymorphism is a fundamental object-oriented design principal in which a virtual method in the base class can be implemented by a derived class using a method with the same name. Any derived class can implement the method in their own unique way. For instance, a vector storing base-class pointers to derived class objects can be used to call all the different implementations of the virtual method in a loop, illustrating polymorphism.

In the timekeeping software example, the manager and IC classes have the same base class, but they also have their own private data. Therefore, the pure virtual method printMemberData is used to demonstrate inheritance by displaying the base class information as well as polymorphism by displaying the private member data of derived class.