

CSC1310: LAB 2

 **(PARTNER Lab – you will work on this lab with an assigned partner)**

**TRAINS**



Concepts

* Classes / Objects
* Class Templates
* Using multiple header files in a program

# DESCRIPTION

You are completing a program that allows for several different types of passenger trains. One train can hold only cats. Another train can hold only wizards. You are going to create a Wizard class, a Cat class, and a Train class. **You are provided the driver, lab2.cpp**. The Train class should be a template class where the type of passenger is the template data type.

# Specifications

## cat class

Attributes:

* name of cat
* breed of cat
* color of cat
* age of cat

Functions:

* constructor – the constructor should have 4 parameters and should initialize the attributes to this sent data.
* overloaded << operator – whenever outside code prints out a Cat object, this function will determine which attributes will be printed and their formatting. You should print out each of the 4 pieces of data on a single line, separated by commas.

Wizard Class

Attributes:

* name of wizard
* age of wizard
* height of wizard

Functions:

* constructor – the constructor should have 3 parameters and should initialize the attributes to this sent data.
* overloaded << operator – whenever outside code prints out a Wizard object, this function will determine which attributes will be printed and their formatting. You should print out each of the 3 pieces of data on a single line, separated by commas.

Train Template Class

Attributes:

* name of the train
* home location of the train
* train passenger capacity (the maximum number of passengers allowed on the train)
* the number of passengers currently on board
* a pointer to a **template** data type. This eventually (in the constructor) will be used to create an array of the passengers.

Functions:

* constructor – the constructor should have 3 parameters and should initialize the train name, home location, and capacity to these values. Then, the constructor should dynamically allocate an array the size of the train capacity. The number of passengers currently on board should be initialized to zero.
* destructor – the destructor should release the dynamically allocated array
* accessor function for the train name (getTrainName)
* addPassenger function – this function should have a template type as a parameter. If the number of passengers on board is euql to the train passenger capacity, your code should print that no more passengers can board the train. Otherwise, your code should place this new passenger in the correct spot in the array and then increment the number of passengers on board.
* printPassengers function – this function should go through the passengers array and print each passenger that is currently on board.

# What to Turn In

Zip all the following files and upload to ilearn.

* lab2.cpp (should not change this file that was given to you)
* Train.h
* Cat.h
* Wizard.h

Sample Output

After you create the three classes required for lab2.cpp, you need to test all the code to make sure you get the same output as below! There is no user input in this program.

**Here are the wizards who have boarded the Hogwarts Express!**

**Bill, 32 years old, 65 inches tall**

**Jill, 24 years old, 45 inches tall**

**Will, 10 years old, 42 inches tall**

**Neal, 88 years old, 73 inches tall**

**Here are the cats who have boarded the Hello Kitty Express!**

**Yella Cat, mixed, orange & yellow, 8**

**Mouth, mixed, black & white, 8**

**Chloe, siamese, black & white, 3**

**Brutus, hairless, skin color, 4**

**Wally, catdog, lime, 1**

**Gladass, siamese, black & white, 6**