

This is part 2 for Exam 1. This part is worth 20 points (part 1 is implemented as a blackboard exam and is worth 80 points).

Please read the instructions carefully. You must put the proper items in the proper file (either `SSandwich.h` or `SSandwich.cpp`. Do **NOT** include any `main` function in your submission. You are encouraged to write your own `main` function to test what you are submitting.

You are submit two files via BlackBoard:

1. A file named `SSandwich.h`, a header file defining the class `SSandwich`. No inline methods are permitted. Any `enum` classes should be defined here also. Your header file should have `#includes` such that `SSandwich.cpp` will compile with only a `#include "SSandwich.h"` as the only `#include`
2. A file named `SSandwich.cpp`, a file that contains the *implementation* of the member functions for the `SSandwich` class.

The sloppy sandwich shop is now accepting take-out orders. Please implement code to support online ordering of sandwiches, with the following properties:

- The `size` may be either `regular` or `glutton`.
- The `breadType` type may be `white`, `wheat`, or `rye`.
- The `breadFormat` may be `wrap`, `sub`, or `sliced`.
- The `contentsList` (represented by a vector of `string`) may include any combination of the following: `Tomato`, `AmericanCheese`, `CheddarCheese`, `Ham`, `Salami`, `Onions`, `Peppers`, `Mustard`, or `Mayonnaise`. (Do **NOT** check the string is a "valid" content for the sandwich. This will allow for custom requests like "only a little mustard" (in addition to saving you having to write some "tedious" code)).

Create a `enum class` for the `size`, `breadType`, and `breadFormat`.

- A. Create the class definition for a class `SloppySandwich` that supports the above and provide **Declarations** for all of the functions that will be defined in parts B-D. (Note: None of the functions in parts B-D should be written as inline functions. All data members must be `private`).
- B. **Define** the following constructors for the `SloppySandwich` class:
  - The default constructor so that the sandwich will have the following properties: `size` is `regular`, `breadType` is `white`, and `breadFormat` is `wrap`.
  - A value constructor that allows the caller to specify all three values for `size`, `breadType`, and `breadFormat`.

In both cases, the `contentsList` initially is empty.

- C. **Define** public accessor methods to obtain (as the appropriate `enum`) `size`, `breadType`, and `breadFormat`, named `getSize`, `getBreadType`, and `getBreadFormat`.

Also **Define** public mutator methods named `setSize`, `setBreadType`, and `setBreadFormat`.

- D. **Define** an overloaded version of the `+` operator so that the user can add contents to the sandwich. The same item may be added more than once.