

# Group Assignment 2: Inheritance

Due on BlackBoard Wednesday 10 October 2018

## 1 Objective

Object-oriented programming is one of the most popular programming paradigms. Moreover, it provides a new abstraction method called inheritance. Inheritance is the capability of a class to derive properties, such as functions and members, from another class. The purpose of this assignment is for students to practice inheritance and function overloading.

## 2 Problem

For this assignment, students will practice inheritance by first creating a class called Food and assigning it some properties. Then, you will create a class called CandyWrapper. Next, you will create a child class called Candy that inherits from Food, adding renewed functionality and having a CandyWrapper attribute.

## 3 Assignment Outline

### 3.1 Create a class named Food

This class will have the following properties:

- A private member of type float called volume
- A protected member of type int called calories
- A protected member of type bool called vegetarian
- A default constructor that sets the *volume* to 0, the calories to 0 and vegetarian to true.
- A constructor that takes a *volume*, *calories* and *vegetarian* as parameters.
- A copy constructor.
- Public accessors and mutators for volume, calories and vegetarian.
- The class overloads the operator ==
- A function called `printItem()` printing the a description of the class in the following format

```
I am a vegetarian Food, my volume is 0.05 fl. Oz and I have 100
calories
```

Or

```
I am not a vegetarian Food, my volume is 0.05 fl. Oz and I have
100 calories
```

### 3.2 Create a class named CandyWrapper

This class will have the following properties:

- A private member of type float called *length*
- A private member of type float called *width*
- A private member of type string called *color*
- A default constructor that sets the *length* to 0, the *width* to 0 and *color* to "NA".
- A constructor that takes a length, width and color as parameters.
- A copy constructor.
- Public accessors and mutators for *length*, *width* and *color*.
- A function called `printItem()` printing the a description of the class in the following format

```
a length of 1 in, a width of 0.7 in and a red color
```

### 3.3 Create a derived class from Food named Candy:

This class will have the following properties:

- A private member of type string called *sweetness*
- A private member of type CandyWrapper called *candyWrap*
- A default constructor that sets all floats and ints to 0 and all strings to "NA"
- A copy constructor.
- A protected member of type float called *color*
- Public accessors and mutators for *color*, *candyWrap* and *sweetness*
- The class overloads `printItem()` printing the a description of the class in the following format:

```
I am a vegetarian Candy, my volume is 0.05 fl. Oz, I have 200
calories, my sweetness level is 0.1, my color is red and my
wrapper has a length of 1 in, a width of 0.7 in and a red color
```

The function `printItem()` from the CandyWrapper class must be used.

- The class overloads the operator `==`

### Submission guidelines:

The submitted code needs to contain the headers and source codes for every class. Please compress the submission as a zip file.

Please use the following main function to test your classes:

```

int main()
{

    Food fd1 = Food(1,200,true);
    Food fd2 = Food(fd1);

    Candy cd1;
    Candy cd2;

    CandyWrapper cw1 = CandyWrapper(1,0.5,"red");
    CandyWrapper cw2 = CandyWrapper(0.5, 0.25, "blue");

    cd1.setVolume(0.05);
    cd1.setCalories(100);
    cd1.setVegetarian(true);
    cd1.setSweetness(0.7);
    cd1.setColor("pink");
    cd1.setCandyWrap(cw1);

    cd2 = cd1;
    cd2.setCandyWrap(cw2);

    if (fd1 == fd2)
        cout << "we are similar !!" << endl << endl;
    else
        cout << "we are different !!" << endl << endl;
    fd1.printItem();
    cout << endl;
    fd2.printItem();
    cout << endl;

    if (cd1 == cd2)
        cout << "we are similar !!" << endl << endl;
    else
        cout << "we are different !!" << endl << endl;
    cd1.printItem();
    cout << endl;
    cd2.printItem();
    cout << endl;

    system("pause");
    return 0;
}

```

**\*\*Please look over in the next section (Sample Output) to see the result of the previous main function\*\***

## 4 Sample Output

```
we are similar !!
```

```
I am a vegetarian Food, my volume is 1 fl.Oz and I have 200 calories
```

```
I am a vegetarian Food, my volume is 1 fl.Oz and I have 200 calories
```

```
we are different !!
```

```
I am a vegetarian Candy, my volume is 0.05 fl.Oz and I have 100
calories, my sweetness level is 0.7, my color is pink and my wrapper
has a length of 1 in, a width of 0.5 in and a red color
```

```
I am a vegetarian Candy, my volume is 0.05 fl.Oz and I have 100
calories, my sweetness level is 0.7, my color is pink and my wrapper
has a length of 0.5 in, a width of 0.25 in and a blue color
```

## 5 Grading

A submitted program that **does not compile** is worth **0 points**.

Your homework is going to be graded on functionality and proper use of inheritance and composition.

Submission requirement: 1 header (.h) and 1 source (.cpp) file for each of the 3 classes.

Grading Criteria	Points
Food class	30
Candy class	30
Candy wrapper class	30
Correct use of inheritance and composition	10