Assignment #5 – OOP / Inheritance

Points: 30

Instructions (3 points)

1. Write a comment at the top that contains the following information
   ```
   // Your Name
   // CS 52
   // Assignment #5
   ```
2. Properly indent, format and comment your code as necessary

Warning: - No late submission is accepted!
- Points will be deducted for entirely commented files!
- Name your files *problem{#}.cpp*, for example *problem1.cpp*, *problem2.cpp*, *etc.*
- *Canvas may append -1, such as problem2-1.cpp which is okay!*

Problem 1 — Shopping Cart — 12 points

Implement a program that tracks customers' purchases. Create the following classes. Use appropriate access modifiers (instance variables should be private!) and data types for each. Don't forget to add getter and setter functions.

- **Item**: This class has the attributes (member variables) called title, description, and price.
  - **Book**: This class inherits from Item. It has an instance variable called pageCount.
  - **Movie**: This class inherits from Item. It has an instance variable called length.
  - **CD**: This class inherits from Item. It has an instance variable called trackCount.
- **ShoppingCart**: This class keeps track of items that were bought. You may limit the number of items in the cart. The cart should have functions to add an item and print the items currently in the cart to the console.
- **Customer**: The customer class stores an id, the first name and the last name and a pointer to a shopping cart object.

Finally, implement a `main` function that creates a customer. Then add one item of each type to the customer's shopping cart and list the items in the cart on the console.

Problem 2 — Pizza — 15 points

Implement a program that creates pizzas based on user order. First, create class called **Ingredient** with one instance variable *description* of type *string* which is set in the constructor and can be get using a getter function. Create the following classes that derive from Ingredient: **TomatoSauce**, **Cheese**, **Dough**, and **Pepperoni**; each with a constructor that passes the *description* argument to the base constructor. Use proper access modifiers!

Create a class **Pizza** which consists of a dynamically allocated array of Ingredient object pointers (a double pointer). The constructor expects the number of maximum ingredients as *int* which is used to dynamically allocate and initialize the array. Implement a destructor that deletes the Ingredient objects and the array. The class has a function *void add(Ingredient* ingredient)* which adds an ingredient to the pizza and a function to print its ingredients to the console.

Create an abstract class **PizzaFactory** with a pure virtual function called *bake()* that returns a *Pizza** object pointer. Create the two subclasses **CheesePizzaFactory** and **PepperoniPizzaFactory** which both explicitly override the *bake()* function. Both functions should remain *virtual*. The *CheesePizzaFactory* instantiates Pizza and adds Dough, TomatoSauce, and Cheese to it. The *PepperoniPizzaFactory* creates a pizza and adds all ingredients including pepperoni to it.

Use the following main method which prompts a user to order a type of pizza and then instantiates the corresponding PizzaFactory for it. After the pizza is created by calling the *bake()* function, the ingredients of the pizza are printed by calling the *listIngredients()* function of the pizza object. Finally, the factory and the pizza objects are deleted.

```
int main()
{
    char choice;
    cout << "Would you like a pepperoni or cheese pizza (c/p)? ";
    cin >> choice;
    PizzaFactory* factory;
    if (choice == 'p')
    {
        factory = new PepperoniPizzaFactory();
    }
    else
    {
        factory = new CheesePizzaFactory();
    }
    Pizza* pizza = factory->create();
    pizza->listIngredients();
    delete pizza;
    delete factory;
    return 0;
}
```

Sample Output:

```
Would you like a pepperoni or cheese pizza (c/p)? p
Pizza with:
Spicy Pepperoni
Chunky Tomato Sauce
Jalapeño Pepper Cheddar Cheese
```