Introduction

This task is about representation of an online shopping website. Some basic elements to represent a website like this are CLOTHES, personal CART and ACCOUNT. You have to code functions, constructors and methods according to this file.

You are provided with three couples of header and .cpp files, and one main file to test your functions, which is seven files in total. All explanations will be provided in this .pdf file as well as example output at the bottom. Please, make sure to follow the explanations and check if every case of a single function is considered.

Let us begin!

Shop

Shop.h file contains two enum classes, which represent type of clothes, and gender for whom those clothes are manufactured, and one usual class, which represent a cloth. Class Shop contains, gender from enum class Gender, cloth_type from enum class Type, price of cloth in floats, size of cloth in integers, brand_name in string and bool available to know if this cloth is available.

Task S.1

Write four different constructors for clothes:

Shop() – default constructor. This constructor should assign insides of class to default values: gender = everygender, type = everytype, price = 0, size = 0, brand_name = “default brand”, available = false.

Other three constructors are with user input. Implement constructor in such a way, that it will assign user input to class and will assign default values to missing inputs. You can check default values above, in default constructor.

Moreover, write output function. For this, overload operator <<.

Task S.2

Write two methods for checking and changing availability of cloth.

check_availability should print that “cloth is available” if available is true and “cloth is not available” otherwise.

Change_availability should assign true/false if available is false/true.

Task S extra

Write boolean operator== that will return true if two clothes are completely identical and false otherwise. We will need it in future.

Write four getters:

1. Get_price – which will return price of a specific cloth
2. Get_gender – which will return for which gender specific cloth was manufactured
3. Get_type – which will return type of cloth
4. Get_availability – which will return bool available of specific cloth

Basket

Basket.h file contains class Cart with three insides:

1. Pointer to Shop – array of clothes
2. Quantity – amount of clothes in array
3. Overall_price – the sum of all prices of clothes

Task B.1

Write default constructor for Cart with default values, where clothes is null pointer, quantity=0 and overall_price = 0.

Moreover, overload operator << for output. It should print all clothes in array.

Task B.2

Overload operator += which will add specific cloth at the end of array of clothes. Please, make sure that all requirements are met, for example, maximum number of clothes in array is not reached.

Task B.3

Overload operator -= which will remove specific cloth from the array of clothes. Please, make sure that all requirements are met, for example, cart is not empty.

Task B.4

Overload operator = which will copy insides of one cart to another cart. Use copy assignment.

Write clear method, which will delete insides of a cart.

Task B.5

Write three filter methods:

1. Filter related to price – will print only those clothes which price is greater than user input price.
2. Filter related to gender – will print clothes for specific gender
3. Filter related to type – will print clothes with specific type

Task B extra

Write one getter: get_quantity – will return number of clothes in an array of clothes.

Please, do not write get_cloth_price getter and buy method for now, we will need them when implementing functions in Account.h. The usage of these two functions will be explained later.

Account

Account.h file contains class Account with five insides:

1. Cash – amount of money in account
2. Name – name of an account
3. Mail – mail of an account
4. Credit_card_number – number of a credit card
5. My_cart – cart with clothes inside. It is located in public for easy access. It is done for simplifying some tasks.

Moreover, it contains two static variables:

1. All_mails – array of all mails ever entered
2. Counter – number of mails in all_mails

Task A.1

Write three constructors:

Account() – default constructor. This constructor should assign default values to insides of a class. Default values are: name = empty string, mail = empty string, cash = 0 and credit_card_number = 0.

Other two constructors are with user input. Implement constructor in such a way, that it will assign user input to class and will assign default values to missing inputs. You can check default values above, in default constructor.

Moreover, overload operator << for output. It should print account name and account mail.

Task A.2

Write two functions void add_mail_to_list and bool check.

1. Add_mail_to_list – appends the all_mails with new mail.
2. Check – checks if a given mail is already in all_mails. Returns true if yes and false otherwise.

Please, make sure that you check all possible cases when implementing these two functions. For example, maximum number of mails is not reached.

Task A.3

Write two methods:

1. Add_cash – adds cash to account's existing money
2. Add_credit_card– adds or changes account's credit card number

Make sure that account exists!

Task A.4

Write two methods:

1. Change_name – changes account name. Make sure that account exists.
2. Change_mail – changes account mail. Make sure that all possible cases are checked, for example, if new mail is already registered.

Task A.5

In this task we are checking the implementation of already existing operator +=, but in account's cart

Task A.6 (!!EXTRA EXTRA!!)

Now we came to the hardest part:

*Method buy with index as input. This method should delete the cloth with specific input and print that purchase was successful.

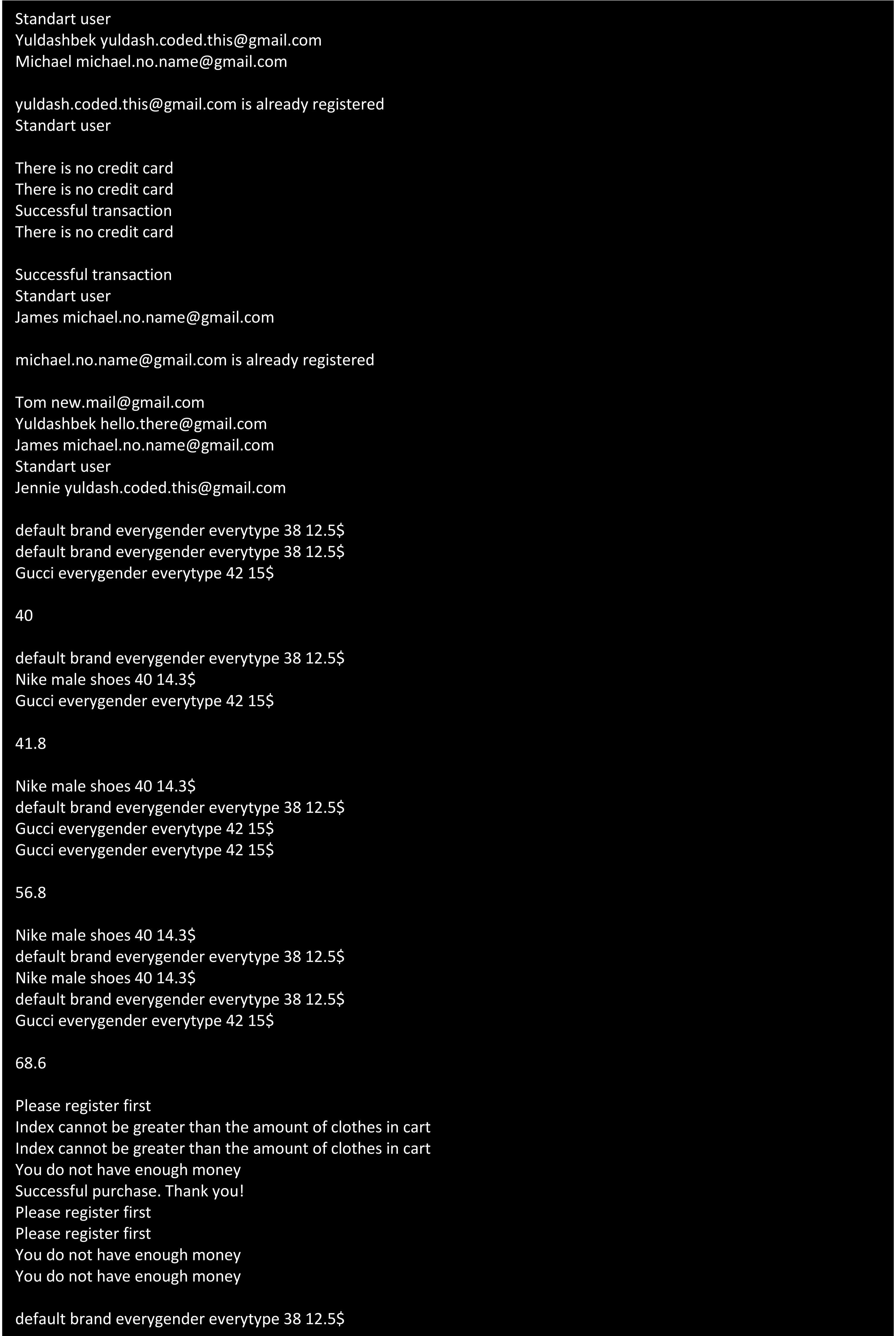
Method buy in account should consider all possible cases and if these cases are not met, should go to method buy in Basket.h file, that we previously skipped.

*get_cloth_price (in Basket.h)– returns price of a specific cloth in array located at specific index. (Hint: you can use previously implemented methods)

*method buy (in Basket.h) - should delete specific cloth in array at specific index. Make sure, that all requirements are met, for example, the cloth is available.

Example output

```
default brand everygender everytype 0 0$
default brand everygender everytype 38 12.5$
Gucci everygender everytype 42 15$
Nike male shoes 40 14.3$

Cloth is currently not available
Cloth is available


Cart is empty

0

default brand everygender everytype 0 0$
default brand everygender everytype 38 12.5$
Gucci everygender everytype 42 15$
Nike male shoes 40 14.3$
default brand everygender everytype 38 12.5$

54.3

default brand everygender everytype 0 0$
Gucci everygender everytype 42 15$
Nike male shoes 40 14.3$
default brand everygender everytype 38 12.5$

41.8

Cart is empty

0

default brand everygender everytype 0 0$
Gucci everygender everytype 42 15$
Nike male shoes 40 14.3$
default brand everygender everytype 38 12.5$

41.8

Cart is empty

0

Gucci everygender everytype 42 15$
Nike male shoes 40 14.3$
default brand everygender everytype 38 12.5$

Nike male shoes 40 14.3$

default brand everygender everytype 0 0$
Gucci everygender everytype 42 15$
default brand everygender everytype 38 12.5$
```



```
Standart user
Yuldashbek yuldash.coded.this@gmail.com
Michael michael.no.name@gmail.com

yuldash.coded.this@gmail.com is already registered
Standart user

There is no credit card
There is no credit card
Successful transaction
There is no credit card

Successful transaction
Standart user
James michael.no.name@gmail.com

michael.no.name@gmail.com is already registered

Tom new.mail@gmail.com
Yuldashbek hello.there@gmail.com
James michael.no.name@gmail.com
Standart user
Jennie yuldash.coded.this@gmail.com

default brand everygender everytype 38 12.5$
default brand everygender everytype 38 12.5$
Gucci everygender everytype 42 15$

40

default brand everygender everytype 38 12.5$
Nike male shoes 40 14.3$
Gucci everygender everytype 42 15$

41.8

Nike male shoes 40 14.3$
default brand everygender everytype 38 12.5$
Gucci everygender everytype 42 15$
Gucci everygender everytype 42 15$

56.8

Nike male shoes 40 14.3$
default brand everygender everytype 38 12.5$
Nike male shoes 40 14.3$
default brand everygender everytype 38 12.5$
Gucci everygender everytype 42 15$

68.6

Please register first
Index cannot be greater than the amount of clothes in cart
Index cannot be greater than the amount of clothes in cart
You do not have enough money
Successful purchase. Thank you!
Please register first
Please register first
You do not have enough money
You do not have enough money

default brand everygender everytype 38 12.5$
default brand everygender everytype 38 12.5$
Gucci everygender everytype 42 15$

40

default brand everygender everytype 38 12.5$
Gucci everygender everytype 42 15$

27.5

Nike male shoes 40 14.3$
default brand everygender everytype 38 12.5$
Gucci everygender everytype 42 15$
Gucci everygender everytype 42 15$

56.8

Nike male shoes 40 14.3$
default brand everygender everytype 38 12.5$
Nike male shoes 40 14.3$
default brand everygender everytype 38 12.5$
Gucci everygender everytype 42 15$

68.6
```

Hope you enjoyed coding!