Assignment 3

1301058

Zhang Junming

Exercise 1

Question

EXERCISE 1 (5 POINTS OUT OF 15)
Design a new class to represent a date like follows: JAN - 25 The class **Date** contains two data members, denoting the 3-letter abbreviation of the month and the date (Note that it is assumed that, for this assignment, Feb. has 28 days.).
Part 1: Basic requirements: 1. Write the constructors, destructor and output function members of this class;

Part 2: Use "operator overloading" to realise the following tasks:
2. Define the assignment operator "=" to assign the value of one Date object to the other;
3. Define the comparator ">" "<" and "==" to compare two dates;
4. Define the unary sign "++" to add one day to a date object:
Eg: Date dToday(MAR, 20); // value of dToday is Mar-20
++dToday; // value of dToday changed to Mar-21

Model Answer

Software Development Process

1. Problem statement

In part 1: Define a class which called Date to write a program, this program could construct Date by letters of month and integer of day. It also could destroy using destructor, the output function should defined in the class member.

In part 2: 3 simple function should run based on the class, one is overloading operator "=" to give the value of private member to a new class; the next is overloading operators "><==" to compare two data which have same class and give the result; the last one is overloading the operator"++" to add one day to given data.

2. Analysis

Inputs:

1) Construct part (the first three letters of a month and a integer which represent the day)
2) Destroy will run automatically after display.
3) In part 2 1. "=" input the value of a new Date
4) In part 2 2."><==" input the value of two new Dates
5) In part 2 3."++" input the value of a new Date

Outputs:

1) Display the date which has full month name with day
2) Display destructor successful
3) Display the new class which receive the new value
4) Display true or false corresponding different compare type(> < and ==)
5) Display next date of given date

Additional requirements or constraint

Jan, Mar, May, Jul, Aug, Oct and Dec have 31 days per month

Apr, Jun, Sep and Nov have 30 days per month

Feb has 28 days per month

3. Design

Algorithm

1. Adding "iostream" and "string" header files.

2. Using of the ste namespace.

3. Define a class which called Date

   Private:
   int month – represent a month
   int day – represent one day

   Public:
   Date() - default constructor
   <1> let month equal to 0(represent January) and day equal to 1(represent the first day in one month)

   void SetDate(string l1, int Day) - a constructor to set date by the first three letters of a month and a integer which represents the day.
   <1> Declare a string l and then store 12 first three letters of the month in it.
   <2> Setting up a loop to confirm which months match the first three letters.
   (a) judge the value of month and day, if user enter correct month and day, give these value to private members.

   ~Date() - a destructor to free memory after using
   <1> Display destructor successful

   void Display() - program will display the full name of the month and day on the screen.

<1> Declare a string Fullname and then store the full name of 12 month in it.

<2> Output the full name of month and day based on the integer in the class member.

<3> If the value of month not in the area between 0 and 11(January to December) or the day not match the correct month (January has 31 days, February has 28 days….) tell user check their input.

Date operator =(Date a) – assign the value of one Date object to the other

<1> let month equal to a.month

<2> let day equal to a.day

<3> Return Date a

bool Date operator >(Date a) – judge the first Date bigger than the second is true or false

<1> if the first month bigger than second month, return true

<2> if the first month equal to the second, judge the day of them, if the day in first month bigger than the second, return true

<3> other condition return false

bool Date operator <(Date a) – judge the first Date smaller than the second is true or false

<1> if the first month smaller than second month, return true

<2> if the first month equal to the second, judge the day of them, if the day in first month smaller than the second, return true

<3> other condition return false

bool Date operator ==(Date a) – judge the first Date equal to the second is true or false

<1> if the first month equal to the second, judge the day of them, if the day in first month equal to the second, return true

<2> other condition return false

Date operator ++() – return the next day of given date

<1> let day plus 1

<2> judge different conditions

(a) If month is Jan, Mar, May, Jul, Aug, Out or Dec, if day bigger than 31, let month plus 1 and day become 1

(b) If month is Apr, Jun, Sep or Nov, if day bigger than 30, let month plus 1 and day become 1

(c) If month is Feb, if day bigger than 28, let month plus 1 and day become 1

(d) If month is Dec, if day bigger than 31, let month become Jan

4. Write the main function

<1> Display the task menu to user.

<2> Using switch and do-while function to store the different operations.

(a) 1- display the task(part 1)

Construct two Date t1 and t2

Display t1

Set t2 is Aug 19 and display it

(b) 2- display the task("=")

Construct Date t3 and t4

Set t4 is June 1 and using operator "=" to give the value to t3

Display t3

(c) 3- display the task(compare)

Let user input two Date and check the information, make sure user enter correct information

Construct two Date t5 and t6

Set Date based on the information which user input

Setting up a loop to let user choose compare in which ways(> < or ==)

Display the result

(d) 4- display the task(add one day)

Let user input two Date and check the information, make sure user enter correct information

Construct Date t7

Set Date based on the information which user input

Display t7

Using operator ++ to add one day

Display t7 again

4. test:

```
1-Part 1
2-Part 2: assign the value of one Date object to the other
3-Part 2: compare two dates
4-Part 2: add one day to a date object
5-Quit
Please choose the task number!
```

Menu

```
Please choose the task number!

1
Write the constructors, destructor and output function members of this class

The date is: January 1th

The date is: August 19th
destructor successful!
destructor successful!
Please choose the task number!
```

Part 1: I set default constructor is Jan 1th and Set anther Date is Aug 19

```
2
assign the value of one Date object to the other
destructor successful!
destructor successful!

The date is: June 1th
destructor successful!
destructor successful!
```

Part 2: "=" I assign the value Jun 1th to new date in the program

```
3
compare two dates
Please enter the first date
the first three letters of month
Jan
integer number of day
2
Please enter the second date
the first three letters of month
Aug
integer number of day
1
1 for >
2 for <
3 for ==
4 over
Please choose the compare ways
1
destructor successful!
false
Please choose the compare ways
2
destructor successful!
true
Please choose the compare ways
3
destructor successful!
false
Please choose the compare ways
4
destructor successful!
destructor successful!
```

Part 2: compare part

```
Please enter the first date
the first three letters of month
january
enter error just three letters
```

```
Please enter the first date
the first three letters of month
Jan
integer number of day
32
enter error
```

When enter wrong month and day

```
4
add one day to a date object
Please enter the date which you want to add one day
the first three letters of month
Jul
integer number of day
23

The date is: July 23th
destructor successful!
destructor successful!

The date is: July 24th
destructor successful!
```

Part 2: add one day ++

```
4
add one day to a date object
Please enter the date which you want to add one day
the first three letters of month
Feb
integer number of day
28

The date is: February 28th
destructor successful!
destructor successful!

The date is: March 1th
destructor successful!
```

```
4
add one day to a date object
Please enter the date which you want to add one day
the first three letters of month
Dec
integer number of day
31

The date is: December 31th
destructor successful!
destructor successful!

The date is: January 1th
destructor successful!
```

Part 2: add one day ++ special condition

Exercise 2

Question

EXERCISE 2 (5 POINTS OUT OF 15)

Write a function

```
bool same_vec (vector<int> a, vector<int> b)
```

that checks whether two vectors have the same elements, ignoring the order and multiplicities. For example, the two vectors {1, 4, 9, 16, 9, 7, 4, 9, 11} and {11, 11, 7, 9, 16, 4, 1} would be considered identical.

Requirements:

1. Get the numbers from keyboard input;
2. The length of the vectors are unknown, it is only determined by the input.

Model Answer

Software Development Process

1. Problem statement

Write a program to construct vector and judge two vectors whether they have the same elements.

2. Analysis

Inputs:

The elements in vector 1 and the elements in vector 2

Outputs:

Whether vector 1 and vector 2 are the same

Additional requirements or constraint

The elements should get from the keyboard and the length should calculate by program; the order and multiplicities of elements are ignored

3. Design

Algorithm

1. Adding "iostream" and "vector" header files.

2. Using of the ste namespace.

3. Define the bool function same_vec(vector<int>a,vector<int>b) – to compare two vector, check whether they are the same

<1> unsigned int n,i,j,t1,t2 – for counting

<2> vector<int>x1 – to store the large vector which user input

<3> vector<int>x2 – to store the small vector which user input

<4> judge the number of elements of two vectors, store the number of elements and these elements in t1 and x1 of the large vector; store the number of elements and these elements in t2 and x2 of the small vector

<5> setting up double loops to compare each element in two vector

(a) if two elements are the same, let j plus 1 and stop compare this element in obj, overloading i to 0

(b) if cannot match this element in obj, overloading i to 0

<6> if all elements in obj match successful return true, else return false

4 Define the input function vector<int> Input_vec(vector<int>&arr) – ask user input the elements of vector

<1> int k = 0 to receive elements

<2> setting up a loop to put each elements in vector and only input 950819 to out the loop

<3> delete the last element in this vector which is 950819

<4> display this vector

5. Write main function

<1> vector<int>a,b – compare these two vectors

<2> Input_vec(a) and (b) using defined function to construct two vectors

<3> using defined function same_vec(a,b) to judge whether they are the same

<4> display the result

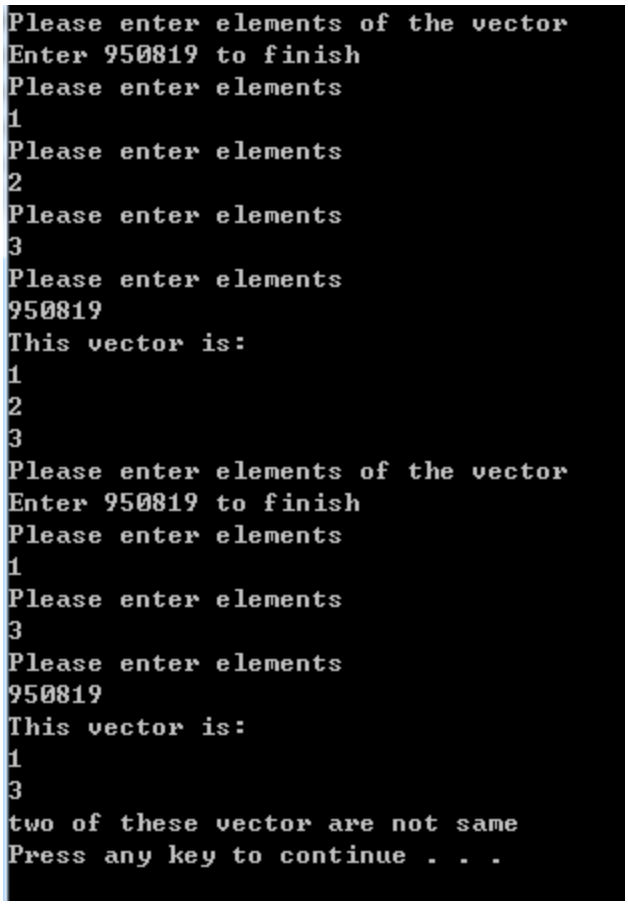
4. Test:

```
Please enter elements of the vector
Enter 950819 to finish
Please enter elements
1
Please enter elements
4
Please enter elements
9
Please enter elements
16
Please enter elements
9
Please enter elements
7
Please enter elements
4
Please enter elements
9
Please enter elements
11
Please enter elements
950819
This vector is:
1
4
9
16
9
7
4
9
11
```

```
Please enter elements of the vector
Enter 950819 to finish
Please enter elements
11
Please enter elements
11
Please enter elements
7
Please enter elements
9
Please enter elements
16
Please enter elements
4
Please enter elements
1
Please enter elements
950819
This vector is:
11
11
7
9
16
4
1
```

```
two of these vector are same
```

Test successful



When enter different vector

Exercise 3

Question

EXERCISE 3 (5 POINTS OUT OF 15)

Write a function to search a character sequence pointed by a pointer (called "obj"), in another character sequence (called "source"). Return the pointer pointing to the found character. If there are more than one target found in source, return the pointer pointing to the first one.

Eg1: search for "C" in "ABCDEF", return the pointer pointing to 'C'.
Eg2: search for "Z" in "ABCDEF", return a NULL pointer.
Eg3: search for "CD" in "ABCDEF", return the pointer pointing to 'C'.
Eg4: search for "CF" in "ABCDEF", return a NULL pointer.
Eg5: search for "A" in "ABCAFC", return the pointer pointing to the first 'A'.

The function header is given by:

```
char *findC (char const *source, char const *obj);
```

Model Answer

Software Development Process

1. **Problem statement**

   Write a program to search an objective string whether in the source string, consider the order, if find return the address of first letter of objective in source string, if not find return null

3. **Analysis**

   Inputs:

   An objective string and a source string

Outputs:

If the objective matches successful, return the pointer pointing to the first letter, if not find, return null

Additional requirements or constraint

This program should consider the order.

3. Design

Algorithm

1. Adding "iostream" and "string" header files.

2. Using of the ste namespace.

3. Define a function which called char *findC(char const *source,char const *obj) – to search objective string in source string

<1> int i,j – for counting

<2> int k – for judge

<3> char *term – to store the result

<4> setting up a loop to count how many letters in objective string

<5> setting up a loop to compare each letters in objective and source string

<6> when only one letter in objective string, if match successful, let k equal to 1 and store the address of the aimed source to term, out the loop in the end

<7> when there is more than one letter in objective string, in the match part, the order of source string should same as objective string

<8> judge the value of k, if k equal to 1, return term, if k equal to 0, return NULL

4. Write main function

<1> string source1 and obj1 to store the letters which user input

<2> ask user to enter the strings of source and objective

<3> convert the string to const char* type using c_str()

<4> using defined function findC.

<5> display the result based on the return value.

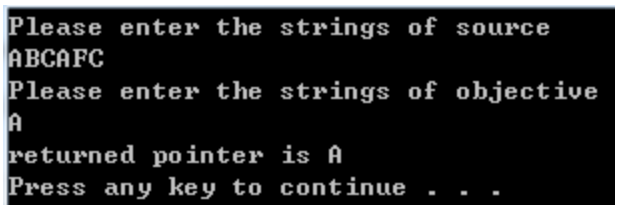
4. Test:

```
Please enter the strings of source
ABCDEF
Please enter the strings of objective
C
returned pointer is C
Press any key to continue . . .
```

```
Please enter the strings of source
ABCDEF
Please enter the strings of objective
Z
NULL
Press any key to continue . . .
```

```
Please enter the strings of source
ABCDEF
Please enter the strings of objective
CD
returned pointer is C
Press any key to continue . . .
```

```
Please enter the strings of source
ABCDEF
Please enter the strings of objective
CF
NULL
Press any key to continue . . .
```




```
if(obj[0] == source[i])// mat
{
    k = 1;
    term = (char*)source[i];/
    break;
```

When match the first letter successful, it will out the loop, therefore it could return the address of first matched letter