In-class Worksheet #6
Fall 2022， Oct 28 and Nov 1, 2022
Computer Science I & Lab

1. Introduction to function: using math functions

http://storm.cis.fordham.edu/zhang/cs1600/CodeExample_Savitch/Chapter04/04-01.cpp

Exercise: add in the main a few lines of code to calculate the following by using mathematics functions (see the slides)

- Floor of 32.35, i.e., rounding down
- Ceiling of 100.01, rounding up to next integer
- 1.2^30 (1.2 raised to power of 30)

2. Simple function that returns a value

http://storm.cis.fordham.edu/zhang/cs1600/CodeExample_Savitch/Chapter04/04-03.cpp

Exercise: Follow the example code to add another function which computes the total cost similar to the total_cost() function, but takes one more parameter which is the tax rate (instead of a fixed 0.05), and call the function in the main, passing a tax rate of 0.08…

3. Declare function before main, and define after main…

Exercise: declare and define the CircleArea() function that we went over in class in this file, and then modify the main so that it calls CircleArea ( ) (instead of unitprice) to calculate the price of the pizze per unit area.

http://storm.cis.fordham.edu/zhang/cs1600/CodeExample_Savitch/Chapter04/04-10.cpp

5. Focus on Coding Style

- Indentation styles support in emacs
    - There are different styles for indenting C++ codes:

      https://www.emacswiki.org/emacs/IndentingC

    - If you press tab, emacs will adjust indentation for the current line automatically based upon the chosen style

- On storm, run the following command to copy emacs configuration file so that Linux style indentation is used by emacs

  cp ~zhang/.emacs ~/.

  Or add this into your .emacs file located in your home directory

```
(setq c-default-style "linux"
          c-basic-offet 4)
```

- Do not put multiple statements in one line, and avoid very long line (break them into multiple lines)
- Use empty lines to separate different parts of the code
- Avoid using ; in if and if/else statement; Avoid using { } and ( ) when it's not needed,
- Simplify your logic when your can:
  - Avoid using ! in Boolean expression
  - Combine conditions together into a Boolean expression
- Avoid duplicative code: through using function, or factoring out common code from if/else statement

Demo on storm: fixing the following code, and use function to check whether a date is valid.

http://storm.cis.fordham.edu/zhang/cs1600/CodeExample_Savitch/style.cpp

BEFORE

```
#include <iostream>
using namespace std;

int main ()
{
  int row = 20;
  int month,day,year;
  int month2,day2,year2;
  char space = '/';

  for (int star = 0; star < row; star++)
    cout << "*";

  cout << endl;

  cout << "Enter date 1 in the format of MM/DD/YYYY: " << endl;
  cin >> month >> space >>day >> space >>year;
  if (!(month>=1 && month<=12))
  {
        cout <<"Invalid date!\n";
        return 0;
  }
```

```
  if ( (day<=31) && (day >=11))
          ;
  else
  {
          cout <<"invalid date\n";
          return 0;
  }
  if (year<0)
  {
          cout <<"Invalid date!\n";
          return 0;  }


cout << "Enter date 2 in the format of MM/DD/YYYY: " << endl;
  cin >> month2 >> space >> day2 >> space >> year2;

    if (!(month2>=1 && month2<=12))
    {
            cout <<"Invalid date!\n";
            return 0;
    }
    if (day2<=31 && day2 >=11)
            ;
    else
    {
            cout <<"invalid date\n";
            return 0;
    }
    if (year2<0)
    {
            cout <<"Invalid date!\n";
            return 0;

    }



    if (month == month2 && day == day2 && year == year2)
    {
            cout << "The two dates are same" << endl;
            return 0;
    }
    if (year2 < year1)
    {
            cout << "date 1 is after date 2" << endl;
      return 0;
    }
  if (year2 > year1)
    {
      cout << "date 1 is before date 2" << endl;
```

```
      return 0;
    }
  if (month2-month < 0)
    {
      cout << "date 1 is after date 2" << endl;
      return 0;
    }

if (month2-month > 0)
    {
      cout << "date 1 is before date 2" << endl;
      return 0;
    }
  if (day2-day < 0)
    {
      cout << "date 1 is after date 2" << endl;
      return 0;
    }
  if (day2-day > 0)
    {
      cout << "date 1 is before date 2" << endl;
      return 0;
    }


  return 0;
}
```

AFTER:

```
#include <iostream>
using namespace std;

int main ()
{
  //indent the body of function
  int month,day,year;
  int month2,day2,year2;

  char space = '/';

  //Display 20 stars
  for (int i = 0; star < 20; star++) //use I,j,k for counter variable
      cout << "*";  //indent the body of the loop
  cout << endl;

  //read in a date and check if it's valid or not
  cout << "Enter date 1 in the format of MM/DD/YYYY: " << endl;
```

```
  cin >> month >> space >>day >> space >>year;
  if (month<1 || month>12 || day > 31 || day <1 || year <0)
  {
          cout <<"Invalid date!\n";
          return 0;
  }



  cout << "Enter date 2 in the format of MM/DD/YYYY: " << endl;
  cin >> month2 >> space >> day2 >> space >> year2;
  if (month2<1 || month2>12 || day2 > 31 || day2 <1 || year2 <0)
  {
          //indent the body of a block statement
          cout <<"Invalid date!\n";
          return 0;
  }

  if (month == month2 && day == day2 && year == year2)
          //indent the yes and else statement of if/else
          cout << "The two dates are same" << endl;

  else if (year > year2 || year==year2 && month < month2 ||
           year==year2 && month==month2 && day<day2)
          //To avoid repeative code, collect all conditions that leads
          //to date 1 is before date 2 together in one Boolean expr.
          //break a long statement into multiple lines
          cout <<"date 1 is before date 2"<<endl;

  else
          cout <<"date1 is after date 2"<<endl;

  return 0;
}
```