**Exercise:**

1. Error handling is a very important concept and a very common issue in programming. Users may not always understand your program and sometimes give you unexpected input value. On Tuesday class we have a program which asking the grade percentage, and display the letter grade to user. The example in class show you how to range the input (1 ~ 100), but sometimes the user will not give you an integer when you asking for in integer input.

Think about this:

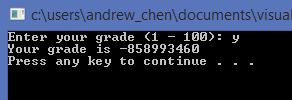
int grade;

cout << “Enter your grade (1 – 100): “ ;

cin >> grade;

cout << "Your grade is " << grade << endl;

In this case, you ask the user to enter an integer. How about user just enter a character ‘y’ ?



Obviously the input ‘y’ is not an integer, so the cin stream can not properly store the value in the integer variable “grade”. C++ gives us an easy way to solve this problem.

Cin.fail() – This boolean function checks your cin stream, if there is a problem cause your cin stream fail, it will return true. Otherwise if the stream works successfully, it return false.

You can try the function by the following steps:

1. Asking user enter an integer input
2. Create a while loop. The condition to terminate the loop is cin.fail() return false (remember cin.fail() return false when the cin stream works fine)
3. If cin.fail() return true, there are two things you have to do in the while loop
   1. cin.clear() //This function will remove the error flag on cin, so cin can work correctly
   2. cin.ignore(INT\_MAX, ‘\n’) //remember the previous exercise in week 1? I just introduce the buffer problem can cause the wrong result from cin, so we can ignore the whole line to avoid the buffer problem. INT\_MAX means the limit of input, You can also use a big number e.g. 10000 to instead. Then we just ignore all the possible wrong input and ready to cin another input.

**Question:**

Create a project which asking user to enter his/her age, then return the year of his/her birthday.

\*Must check the valid input for integer only.

Sample output:

Next page

