Practice 1

Objective – Getting familiar with MS visual studio

1. Run the visual studio and look around
2. Go through a complete build (compile + link) process
3. Examine compile errors
4. Practice the debugger in the visual studio
5. Get familiar with cout, cin, and variables through simple programing problems

Problem 1: Hello World!

- Run the visual studio and open an empty solution
  * Refer to the supplementary material "supplementary_Visual_studio.pdf" enclosed
- Type in the following simple program code and build it
- Watch what happens

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

int main()
{
        cout << "Hello, world" << endl;
        cout << "I am [your name]" << endl;
        return 0;
}
```

Think about why we need those indentations here

Problem 2: Compile errors

- Type in the following program code segment and try to compile it
- Inspect the compiling results, and double-click the errors
- Understand how compile errors are reported in the Visual Studio
- Find out why the compiling was failed
- Fix the program

```
include <iostream>
using namespace std;

int main
{
        cout << 'We are learning ';
        cout << 'C++ language';
        return 0;
}
```

Problem 3: Using a debugger

- Debugging is extremely useful when you encounter a bug (defection of a program).
- Read through the debug section of the supplementary materials and understand why and when we need it.
- Type in the simple program code (next page), and build it as a debug mode.
- Set a breakpoint (F9) at one of the couts, and run the program as debug mode (F5).
- See how your code are executed line by line when you press the step over command (F10).

* At this moment, debugging is not quite useful for you but will be your good friend later when your program is getting more complicated and pointers are involved. More details, tutorials, and lectures on debugging can be found in the internet. Google "Visual Studio debugger."

e.g.,

http://www.munsam.info/xe/?module=file&act=procFileDownload&file_srl=282297&sid=29a40a97713fe36c36612193552f8b7c

http://www.codeproject.com/Articles/79508/Mastering-Debugging-in-Visual-Studio-2010-A-Beginn

http://chocodonut.tistory.com/408

http://home.sogang.ac.kr/sites/mics/testmenu6/Lists/b12/Attachments/33/lecture%2011%20-%20c%20-%20debugger.pdf

Problem 3

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

int main()
{
    cout << "*" << endl;
    cout << "**" << endl;
    cout << "***" << endl;
    cout << "****" << endl;
    cout << "*****" << endl;
    cout << "******" << endl;
    cout << "*******" << endl;
    cout << "********" << endl;
    cout << "*********" << endl;
    cout << "**********" << endl;
    return 0;
}
```

Set a breakpoint here

and execute the program line by line

Problem 4: Using comments, variables, and constants

- Type in the Program 2-3 in the textbook and build it.
- Check the results.
- Insert your comments at each line explaining the meaning of it.
- Add a constant and a variable representing the Euler's number (2.71828). Make your own identifier for the number, and print the constant and variable of it using cout.

Program 2-3 Memory constants

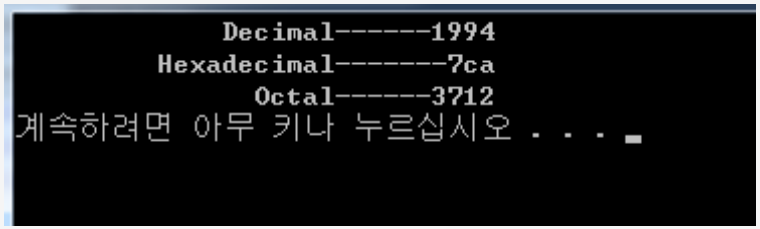
```
1  /* This program demonstrates three ways to use constants.
2        Written by:
3        Date:
4  */
5  #include <iostream>
6  using namespace std;
7
8  #define PI 3.1415926536
9
10 int main ()
11 {
12    const double pi = 3.1415926536;
13
14    cout << "Defined constant PI: " << PI << '\n';
15    cout << "Memory constant pi:  " << pi << '\n';
16    cout << "Literal constant:     " << 3.1415926536 << '\n';
17    return 0;
18 }  // main
```

```
Results:
Defined constant PI: 3.14159
Memory constant pi:  3.14159
Literal constant:    3.14159
```

Problem 5: Formatting in cout

- Write your own program that print out the following formatted table (note the vertically aligned numbers and words).
- Your program should use setw(), setfill(), hex, and oct.
- Refer to Program 2-5, 2-6, and 2-7.

Output result



Problem 6: usage of cin and variables

- Write a program that gets three integers from user input and then print vertically (each in one line), first forward and then reversed (the last one first), as shown below.
- Define three integer variables for storing the inputs.
- Print the variables using cout

Output result

```
Please enter three numbers: 20 30 40
Your numbers forward:
20
30
40
Your number reversed:
40
30
20
```

Problem 7: usage of cin, cout, and variables

- Write a program that gets 6 integers from user input and then print them three in a line separated by comma as shown below.
- Note the formatted output.

Output result

```
Please enter 6 integers: 1 100 50 20 500 2
Input:
1 100 50 20 500 2
Output:
  1, 100,  50
 20, 500,   2
```