

EXPERIMENT 1

MICROSOFT VISUAL STUDIO AND C PROGRAMMING

Aims

1. Learning primary functions of Microsoft Visual Studio
2. Introduction to C Programming
3. Running C programs using Microsoft Visual Studio

In this experiment we will be using Microsoft Visual Studio 2008; however, with small differences, usage of most integrated development environments are similar.

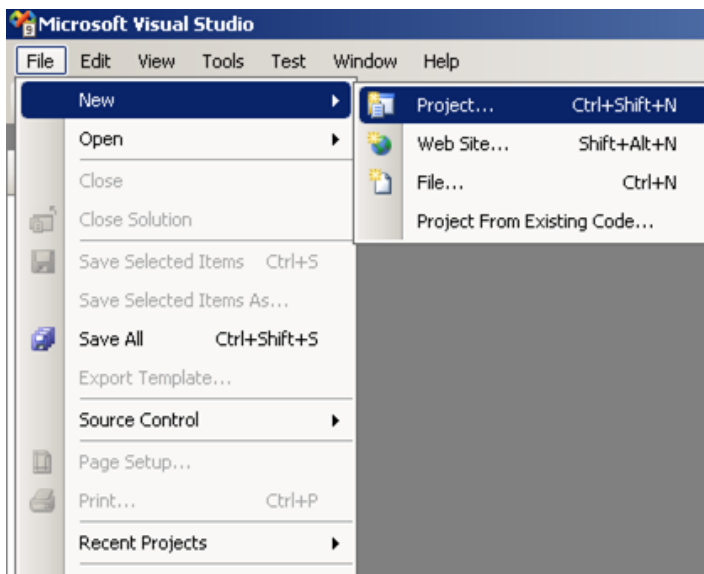


Figure 1: File menu

Creating a New Project

To create a new project, start Visual Studio 2008 (Start → Programs → Microsoft Visual Studio 2008 → Microsoft Visual Studio 2008). Select Project from the New list contained in the File menu, as shown in Figure 1. This operation will display a dialog screen shown in Figure 2. Select “Visual C++” from the “Project types” list shown on the left. Select “Empty project” from the right list and give a name to your project. (Note that you may need your

project later on; therefore, choose a suitable name that you can remember). These steps will create an empty project for you to work on.

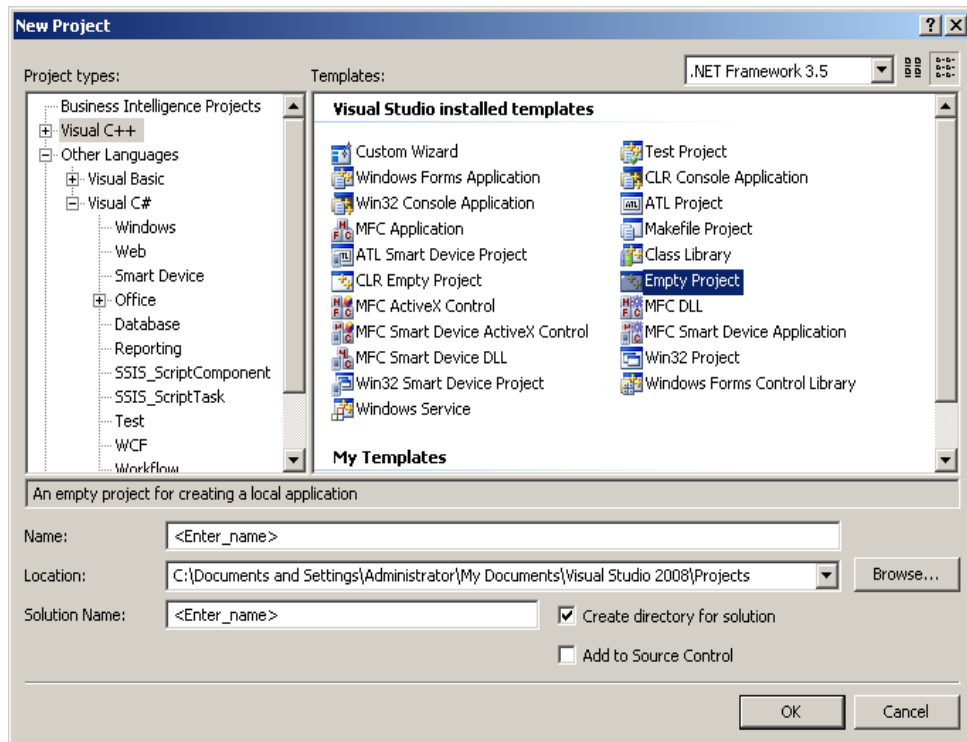


Figure 2: New project window

The first step to do in a new project is to create a file that we can write our codes into. Firstly, find “Project explorer” window. If it is hidden, you can open it from “View” menu. Right click on empty part of Project explorer and choose “New item” from “Add” menu as shown in Figure 3. This operation will open the window shown in Figure 4. From this window, choose “C++ file” and supply its name to create your file.

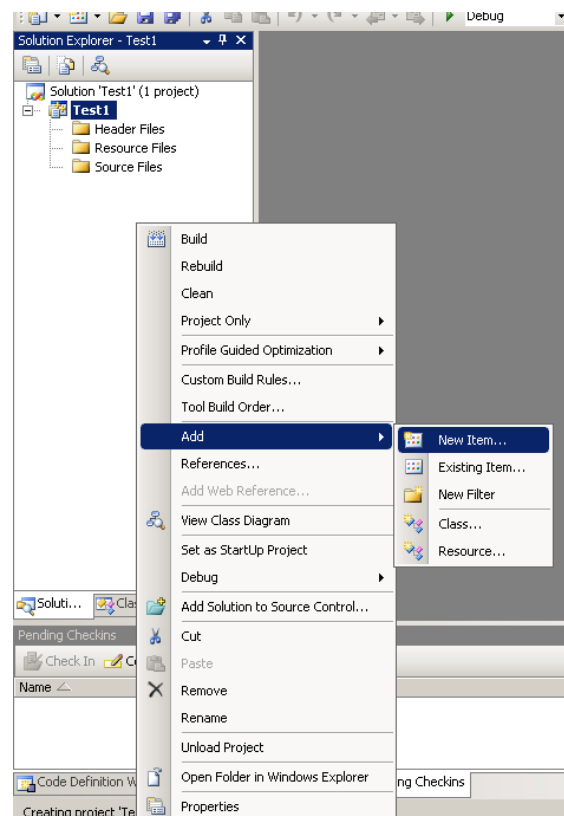


Figure 3: Add file menu

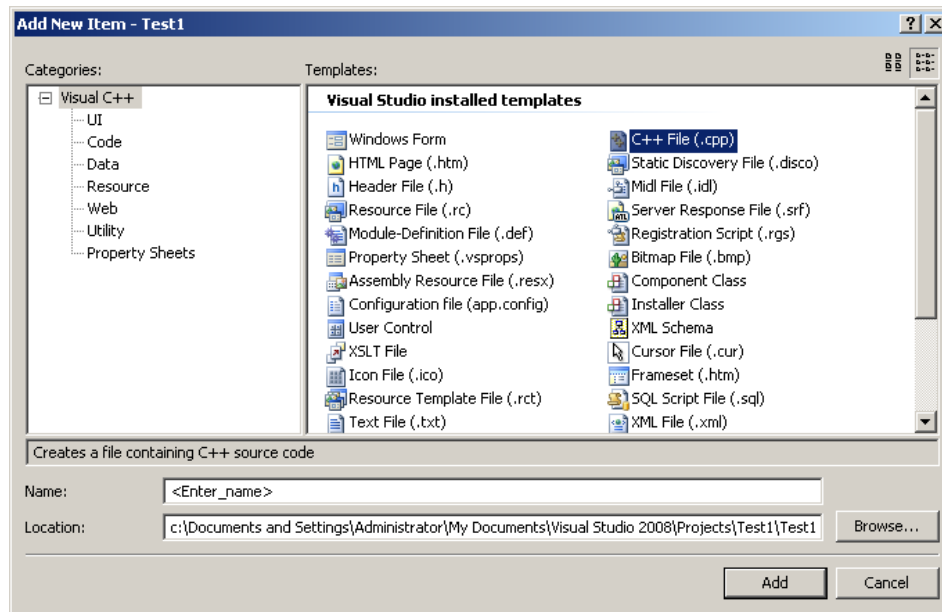


Figure 4: New item dialog

First C Program

Note: Performing these operations as you read them will help you to understand it better.

The first part that should exist in a C program is its entry function. This function should be named as “main”, so that C compiler will know which method it has to call. This function should return an integer (int) after its operation. For now we will only use the value 0, which denotes that the application completed its task as expected. The following is the shortest C program that can be executed.

```
int main() {
    return 0;
}
```

As a simple program we can use “Hello World!” program which will print “Hello World!” to the screen and exits. To perform this operation we will require standard input/output library. We can perform include operation with the following preprocessor directive. This command should be written in global scope, which resides outside the function and other structure definitions. Moreover, the preprocessor directives, which starts with # symbol, does not need semicolon (“;”) afterward. Generally, preprocessor directives are place at the top of your file.

```
#include <stdio.h>
```

After adding this library file to our project, we can use **printf** (print formatted) function to print text to the screen. This function is defined in “stdio.h” file. The following code fragment will print “Hello World!” to console screen. This code should be inside our main function. However, since our

program ends at **return 0;** line, printing code should be before that line. The “\n” added to the end of the text denotes that the text coming afterward will be printed on the next line.

```
printf("Hello World!\n");
```

In Visual Studio 2008, the console application that we write automatically closes after they are finished. Because of this, the result will not be available for us to read. We will use “pause” command from MS-DOS to prevent this. We can use “system” command to run programs from the operating system. However, we will need to include related library, which is standard library (stdlib.h). Therefore, following line should be added to our program. As before, it should be on top.

```
#include <stdlib.h>
```

The following command runs the pause program which will wait for user to press a key.

```
system("pause");
```

This program stops our program execution until a key is pressed. Therefore, it should be after the operation we perform. However, it should be before the end of our program, otherwise it will not be effective. You can use this command to pause your program temporarily on other occasions too.

The following is the whole program.

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    printf("Hello World!\n");

    system("pause");

    return 0;
}
```

Running Your Program

To run your program you can press the button shown in Figure 5 or “F5” key. After issuing run command, the window shown in Figure 5 will be displayed. This dialog window asks if we want to compile our program. Since we want to run the codes we typed we have to compile our program. Therefore, the answer should be “Yes”. If there are no errors in your program, a console screen will be displayed. If your program requires input, you can enter it through this screen. Also your program output will be shown in here.

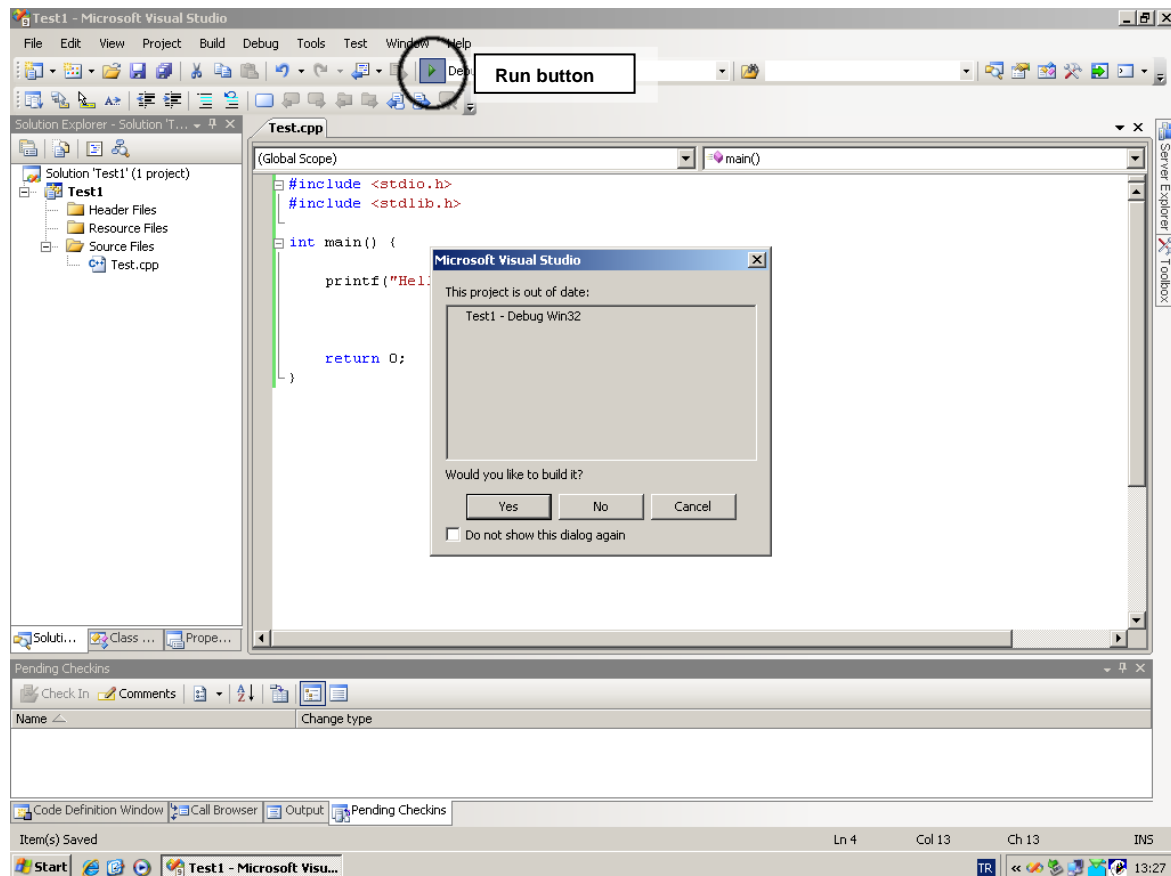


Figure 5: Running your program

Experiments:

- 1) Write a C program to take an integer number from the keyboard and then prints if the number is odd or even.

Note: 2, 4, 6, 8, 10, are even numbers and 1, 3, 5, 7, 9, 11, are odd numbers.

- 2) Write a C program to find the sum of first n numbers where n is entered by user.

Note: if n is equal to 4 then the sum of 1,2,3,4 is 10. You should use loop to write this code.

- 3) Write a C program to store information of 5 employers using a structure. You should create a structure (employers) to contain name, age and salary of each employer as its data member. Then, an array of structure of 5 elements should be created. Then, data (name, age and salary) for 5 elements is asked to user and stored in array of structure. Finally, the data entered by user is printed.

Note: you can define an array of stucture as: **struct employers emp[5];**

- 4) Write a C program to add two numbers using pointers.

Note: You should save the address of two numbers into two pointers and then add them.

- 5) Write a C program to read two 1-D array of size 3 from the keyboard and then using a function copy elements of first array into second array.

Note: use call by value and call by reference to print the array elements. Try to understand the difference between call by value and call by reference.