1 Introduction to Programming

1.1 What Does It Mean to Program

1.1.1 Algorithm

Programming is not only coding. Primarily it implies structuring of the solution to a problem and then refine the solution step by step. When refined to a level deep enough, you have created an **algorithm**. Then it is time to translate each step of the algorithm to program code.

Suppose you have a problem that needs to be solved. Then you begin with writing a sequence of operations at an overview level that need to be performed to solve the problem. Then you start from the beginning again and focus on one operation at a time and find out whether the operation needs to be refined to more detailed steps. Then you proceed to the next level and refine further. This refinement process goes on until you arrive at a level deep enough to start coding.

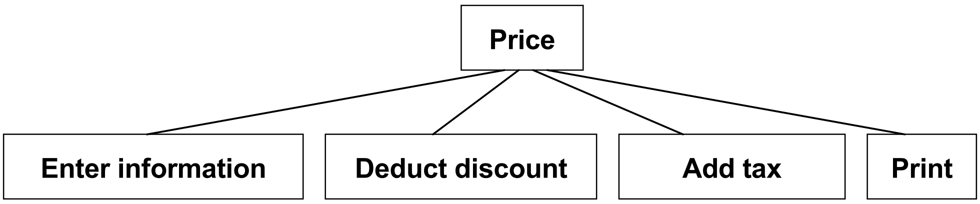
Creating an algorithm to solve a problem is in general the most laboursome task of the programming work. Many people do the mistake of starting to code at once, which makes you focus on code details and forget the actual problem to be solved. That gives an unstructured and inefficient code hard to understand and maintain.

That's why we emphasize that you structure your logic train of thought and construct a good algorithm before starting to code.

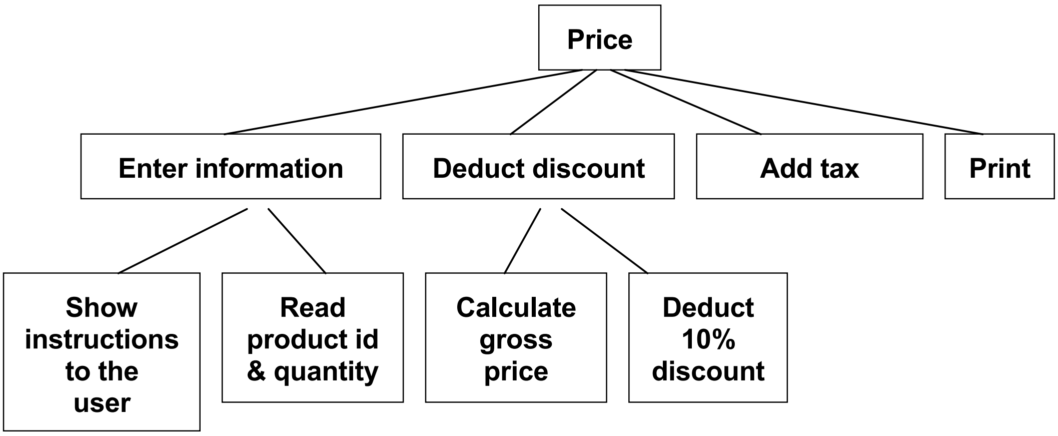
1.1.2 JSP

A JSP graph is a tool to create an algorithm. JSP is an abbreviation for Jackson Structured Programming and is a commonly used instrument for logic structuring. Let's take an example.

You are supposed to create a program that calculates the price of a product to be bought by a customer. The customer specifies the product id and the requested quantity. The program should then calculate the relevant discount, add tax and show the customer price. A JSP graph could look as follows:

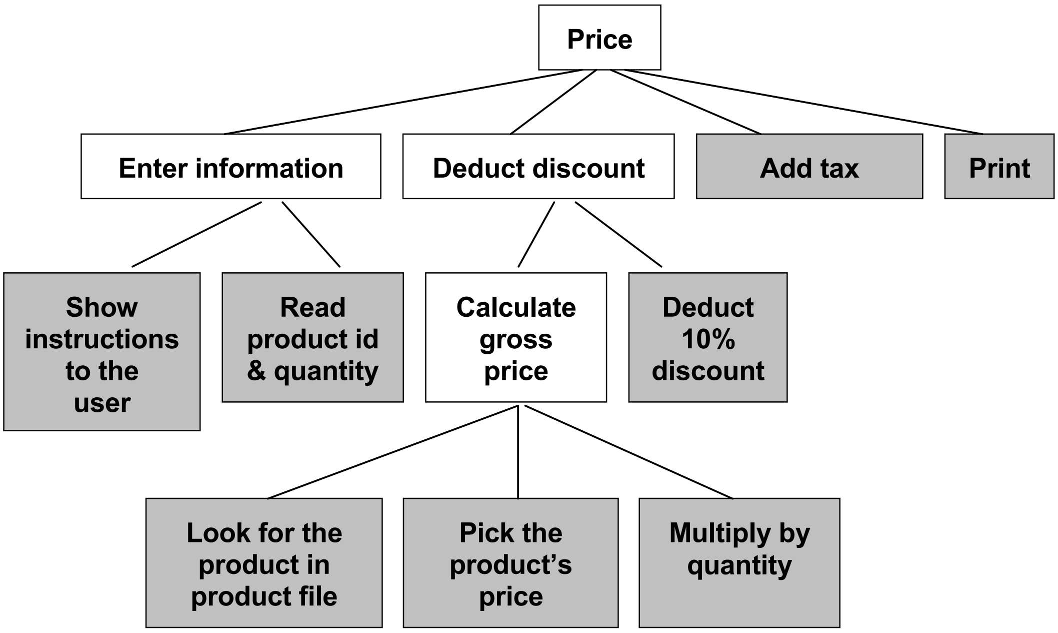


The upper box is the name of the program. We have split the solution into four steps at an overview level. You read the steps from left to right. As you probably realize the algorithm is too rough to be able to write code. So we proceed by refining the solution to the next level:



The box ”Enter information” has been split into two steps, “Show instructions to the user” and “Read product id & quantity”. In the same way we have refined the box “Deduct discount”.

We could break down the box ”Calculate gross price” further:



We could refine the algorithm further, but let us say that we are satisfied with the detail level. The shadowed boxes in the graph are the end points at the lowest level, the “leaves of the tree”. These are the boxes to be used for coding, from left to right.

We will work a great deal with JSP graphs in the program examples of the course.

1.1.3 Sequence - Selection - Iteration

Each program is logically built up from three basic logic principles:

- **Sequence** – the program performs instructions in sequence, one after the other.

- **Selection** – the program selects one of several operations depending on the prerequisites. The program thus makes a selection based on some condition.
- **Iteration** – the program repeats a series of instructions a certain number of times.

The logic principles can also be combined. For instance, a sequence of instructions can be repeated a number of times if a specific condition is satisfied, otherwise another sequence of instructions should be performed a specific number of times.

All programming languages use these three logic principles. If you have built up your algorithm in a correct way, it is only a question of selecting a programming language when the coding is to take place. The price calculation algorithm above should consequently give the same result irrespective of whether the code is written in C++, Java or VisualBasic.

In the JSP graph above the box "Calculate gross price" is refined in a *sequence* of three operations, from left to right:

- Look for the product in product file
- Pick the product's price
- Multiply by quantity

The box ”Look for the product in product file" could suggest an *iteration*, e.g. "Read next product id until we find the product id stated by the user".

The discount calculation in the price program above could imply a more differentiated discount situation:

- If the gross price is between 100:- and 500:- the customer will get 5% discount.
- If the gross price is between 500:- and 1000:- the customer will get 8% discount.
- If the gross price is above 1000:- the customer will get 10% discount.

Here the program must do a *selection*.

1.2 Coding

When you have refined your algorithm to a level detailed enough, it is time to write code. This written code is called **source code**. The code must of course follow the rules in effect for the programming language in question, it must follow the syntax. Each programming language has its own rules.

You can in principle use any word processor or text editor you like, such as the program Notepad, Wordpad or Word. If you use word processors like Wordpad or Word, you must save the file as pure text file (.txt).

It is however recommended to use the text editor present in the program development package you are using. The advantage is that you will get some support when coding. Microsoft Visual C++, which is the program development package used in this course, contains an editor which:

- Shows key words in C++ in blue colour and comments in green colour,
- Provides IntelliSense, i.e. proposes code alternatives in certain situations,
- Supports context sensitive help, i.e. shows an explanation of a certain code item if you put the cursor on the item and press F1,
- Provides extensive support at debugging by allowing you to execute the code up to a certain breakpoint, where you can examine variable values at this particular position.

There are other development tools for C++ like Borland and Dev C++. The tools differ somewhat as concerns small details of the code. You can use any tool, the important thing is that you learn to "think" structured programming. In this course we have used Microsoft Visual C++ 2008, and all program examples are tested in this environment.

C++ is a very extensive language that can be used both within basic structured programming and object oriented programming. Furthermore, windows programming in the graphic Windows environment is supported. C++ has, thanks to its level of detail, its strength in digging deeply into the most obscure corners of the computer, control the operating system, communicate with hardware, circuit boards and external equipment like measurement units and communication devices. In this course, however, we will stick to basic structured programming.

C++ is a compact language with many symbols having their own meanings. This means that C++ code looks complicated to the novice. On the other hand it provides many tools for efficient coding. Programs written in C++ are very rapid due to the fact that the compiler optimizes them to each specific processor type. That is the reason why you mostly use C++ in situations where processor time and performance are ultimate.

1.3 Compiling and linking

When you have written the code of your program, it should be compiled, i.e. translate it to machine code consisting of 1's and 0's. That is the level understood by the processor. Before reaching this level, there is an interim level of code, called object code. Thus the compilation is made in two steps, first from source code to object code, and then from object code to machine code. In Microsoft Visual C++ you don't have to bother about these two steps, because the compilation from source code to machine code is taken care of by just clicking a button.

When writing a program, you often split the code into several files of source code. The different files contain references to each other. When all source code has been compiled the different parts of the program must be linked together to one single executable program (exe file). In some environments the linking must be initiated manually. In Visual C++ the linking is taken care of automatically directly after the compilation.

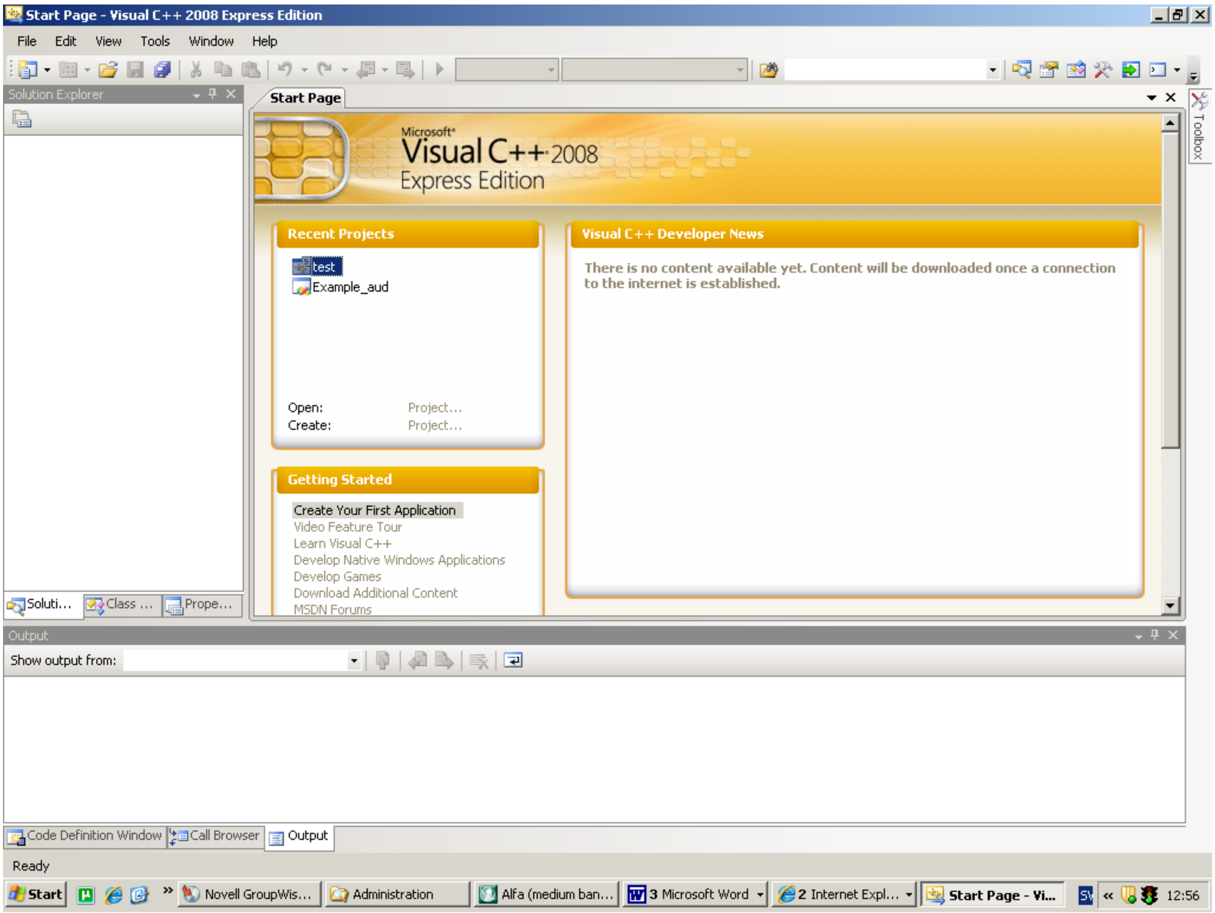
1.4 The First Steps with Visual C++

We will start with creating a program that prints 'Hello World' on the screen.

Click the Start button and select:

1.4.1 All programs – Visual C++ 2008 Express Edition -Microsoft Visual C++ 2008 Express Edition

The Visual C++ Start Panel is displayed:

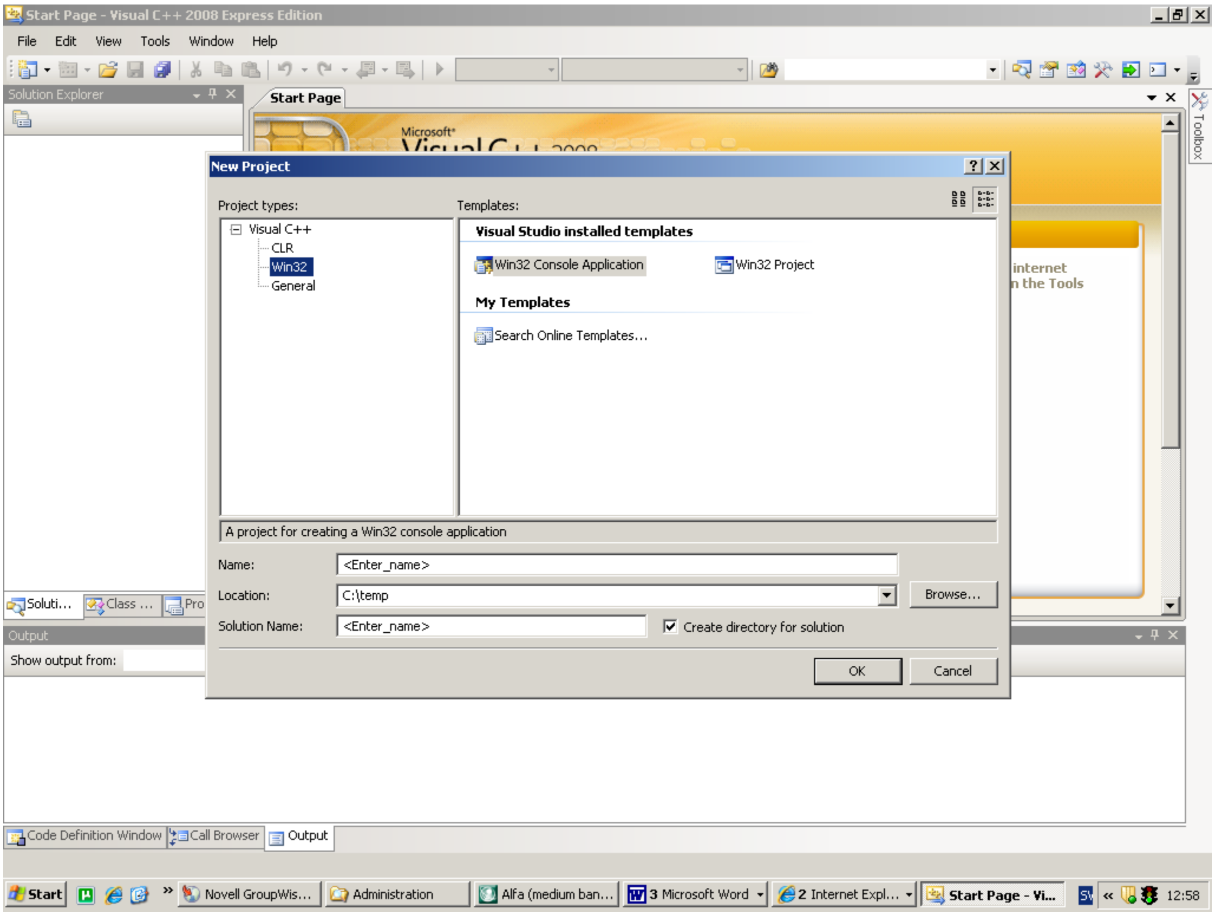


There is no content available yet. Content will be downloaded once a connection to the internet is established.

To create a new program, select from the menu:

1.4.2 File – New – New project

A window is displayed:



New Project

Project types:

- Visual C++
  - CLR
  - Win32
  - General

Templates:

Visual Studio installed templates

Win32 Console Application   Win32 Project

My Templates

Search Online Templates...

A project for creating a Win32 console application

Name: <Enter_name>

Location: C:\temp

Solution Name: <Enter_name>

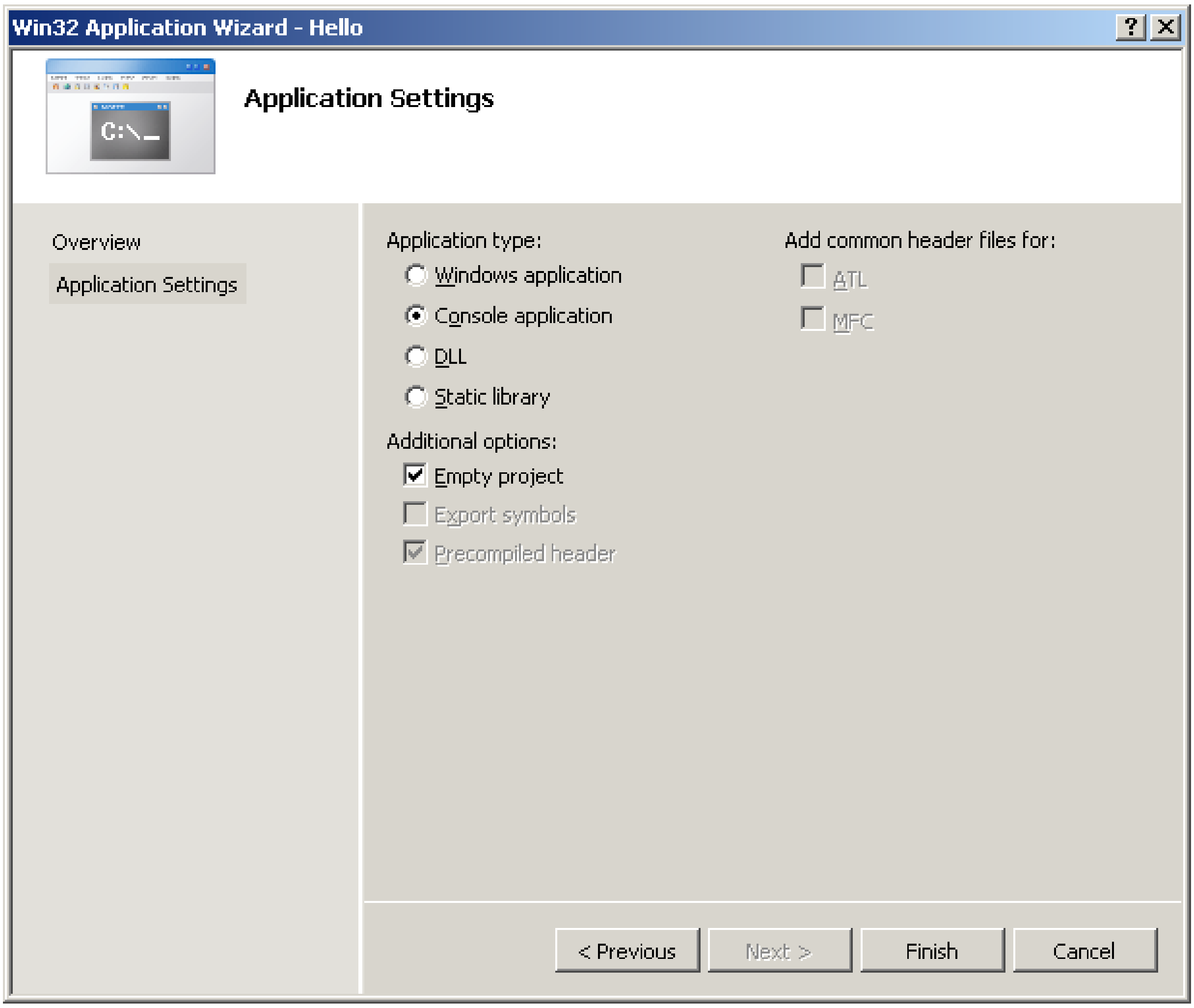
☑ Create directory for solution

Select as shown by the window above.

1.4.3 Win32 – Win32 Console Application

Also specify the name of the program, for instance Hello, in the box after 'Name', and indicate a suitable folder with the 'Browse' button, where the programme is to be stored. A particular sub-folder will be created with that name.

Click 'OK'. A new window will be displayed:



Win32 Application Wizard - Hello

Application Settings

Application type:

○ Windows application

● Console application

○ DLL

○ Static library

Additional options:

☑ Empty project

□ Export symbols

☑ Precompiled header

Add common header files for:

□ ATL

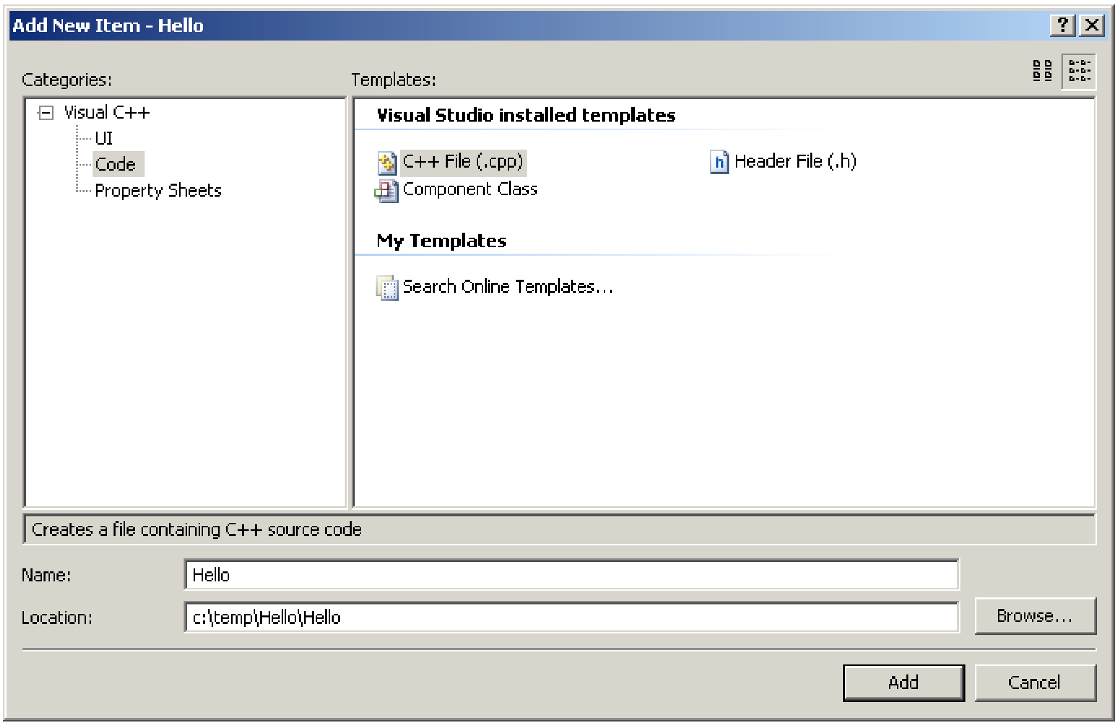
□ MFC

Click on 'Application Settings', check the box 'Empty project' and click on 'Finish'.

The project is now created but contains no code files yet. Add one by selecting from the menu:

1.4.4 Project – Add New Item

A window is displayed:



Creates a file containing C++ source code

Name: Hello

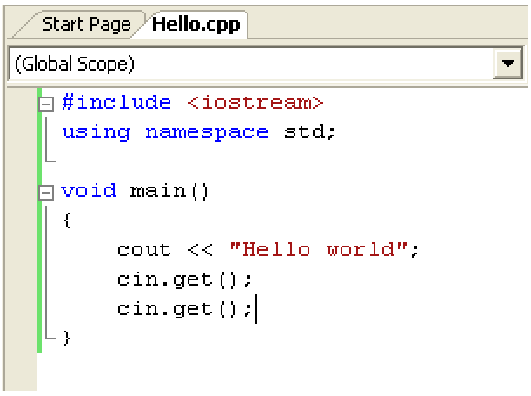
Location: c:\temp\Hello\Hello

Select ’**Code**’ in the left part and **‘C++ File (cpp)’** in the list of icons.

Enter a name of the file to be created in the box after ’Name’.

Click ’Add’.

A code window is displayed where you can enter code:



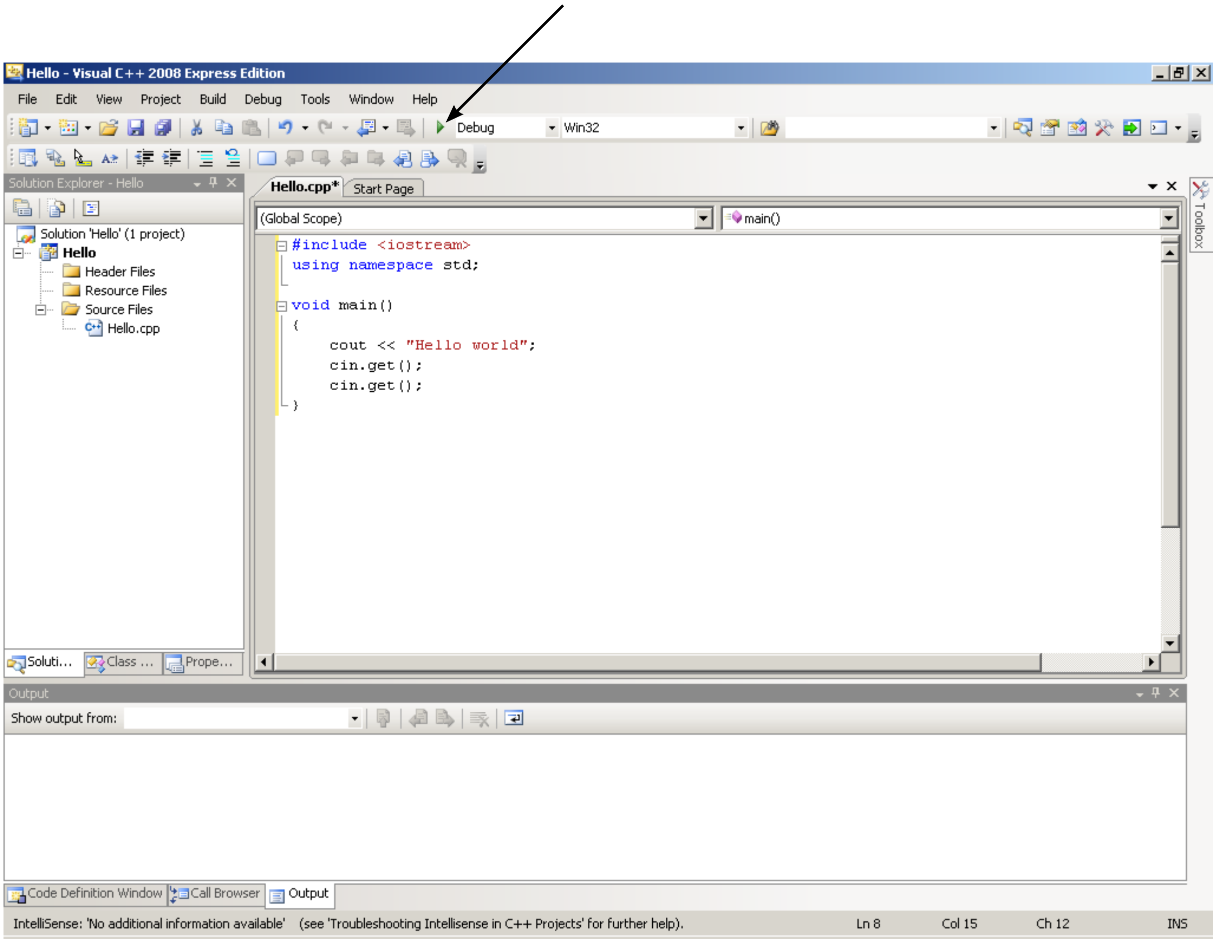
```
#include <iostream>
using namespace std;

void main()
{
      cout << "Hello world";
      cin.get();
      cin.get();
}
```

We will not explain all details in this program. That is done in the next chapter. The main thing is that you get started with the system and are able to write and compile code and run programs.

The changes necessary in the code compared to previous versions of Visual C++ are:

- Do not use .h in include statements, it should be:
  `#include <iostream>`
  However, in some include statements the .h should be kept
- Insert the statement
  `using namespace std;`
  which indicates to the system where the standard library is
- Insert the following statement as the last statement in your programs:
  `cin.get();`
  which makes the program stop and you will get an opportunity to view the console window with the displayed output. Sometimes two cin.get() –statements are needed:
  ```
  cin.get();
  cin.get();
  ```

Compile and run the program by clicking the arrow icon:



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

void main()
{
     cout << "Hello world";
     cin.get();
     cin.get();
}
```

IntelliSense: 'No additional information available' (see 'Troubleshooting Intellisense in C++ Projects' for further help).

When you see the output in the console window you can stop the program by pressing Enter once or twice. It is the cin.get () –statements at the end of the program that waits for this Enter press.

We will create another program that asks the user for the unit price of a product and the quantity, and then calculates the total price.

Before you create a new program, you should close the project of the previous program, otherwise it will trouble the new program Select:

File - Close Solution

and answer 'Yes' to the question of closing all windows.

One thing you should remember is that each program takes 5-6 Mbyte disk space, due to that a lot of extra files and a Debug folder is created, which is necessary for using the debug function. These extra files and the Debug folder is recommended to be deleted after completion of a program, otherwise you might soon run out of disk space. The only thing to be saved is the cpp file where your source code is stored, which in our example has the name Hello.cpp.

Start a new program like in the Hello example and enter the following code:

```
#include <iostream.h>
void main()
{
     int iNo;
     double dblPrice, dblTotal;
```

```
      cout << "Enter price per unit ";
      cin >> dblPrice;
      cout << "Enter quantity ";
      cin >> iNo;
      dblTotal = dblPrice * iNo;
      cout << "The total price is " <<dblTotal<< endl;
  }
```

We will not explain all details in this program. That is done in the next chapter. We will only touch the main steps of the program.

After the 'void main' line there are two lines where we declare some variables needed for storing of entered and calculated values.

The cout-line outputs a text to the screen. The subsequent cin-line makes the program stop and wait for the user to enter the price per unit and press Enter. This is repeated for the quantity of the product.

The dblTotal-line calculates the total price by multiplying the unit price by the quantity.

The last line outputs the total price.

When you compile the program it might happen that you have typed wrong. Then you will get a list of compilation errors in the window at the bottom of the screen. Double-click the first error to make Visual C++ indicate the erroneous line. Correct the error. Go on with the other errors and compile again. You might need to recompile a number of times.

Finally, when all errors have been corrected, run the program by clicking on the exclamation character button.

1.4.5 Where Is the Program

You can in Explorer examine the folder where you saved your program. During the compilation a sub-folder is automatically created called Debug. If you open it you will find the exe-file. You can now run the program by double-clicking the exe-file.

You can also put your program at the start menu. This is preferably done with the drag and drop method. Push the mouse button on the exe-file and hold it down, draw the mouse pointer to the Start button and release the mouse button. When you then click the Start button you will find your program on the Start menu.