SPM 2102 PROGRAMMING LANGUAGE 1

Introduction to C++ (Environment and data type) Part 1

By

NORAH MD NOOR &

MEGAT AMAN ZAHIRI MEGAT ZAKARIA

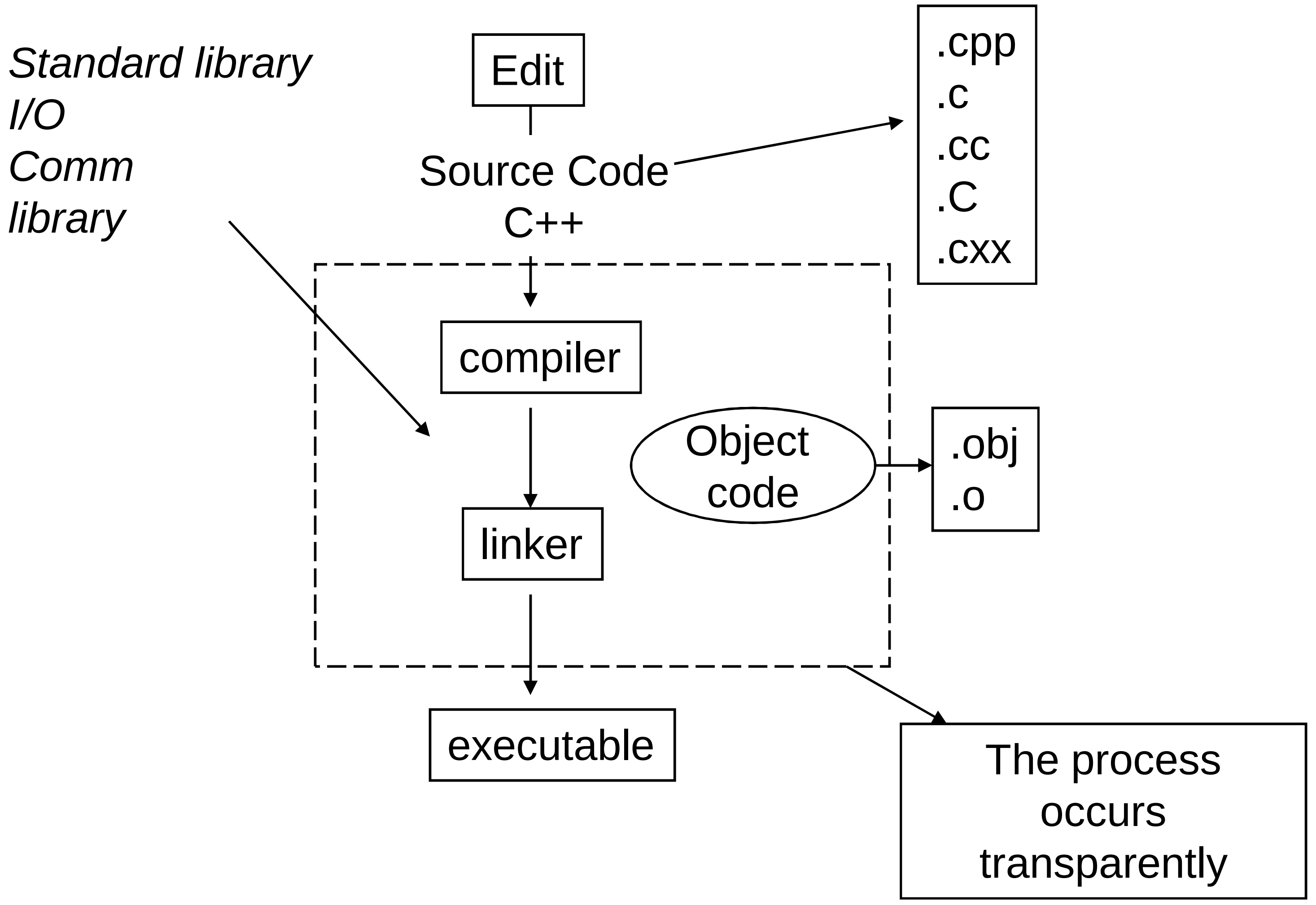
At the end of this lecture, you should learn:

- Environment of C++ programming
- Structure of C++ programming
- C++ data types
- Elements of C++

- C++ was created on 1979 by Bjarne Stroustrup at the Bell Laboratories, New Jersey – 10 years after the ‘birth’ of C language
- C++ contains all of C elements with some additional features – with the purpose of eliminating the flaws that exists in C
- C emphasize on structured programming while C++ is rather more object oriented programming.

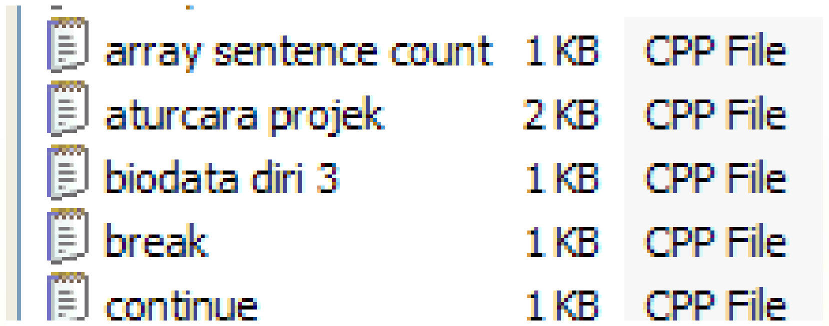
- A more massive and complex application could be achieved with this object oriented method of programming (C++).
- The standard version of C had been released on the year 1989 - ANSI C (*American National Standard Institute*)

- C and C++ programs were produced in text files (.txt) using text editing applications - e.g: Notepad, vi, emacs, pico etc
- Programs that were produced in this form are known as source code
- Source codes that have been compiled will produce object codes and later will converted into .exe by a linker
- Object code is a machine code that is not complete


Standard library
I/O
Comm
library
Edit
Source Code
C++
.cpp
.c
.cc
.C
.cxx
compiler
Object
code
.obj
.o
linker
executable
The process
occurs
transparently

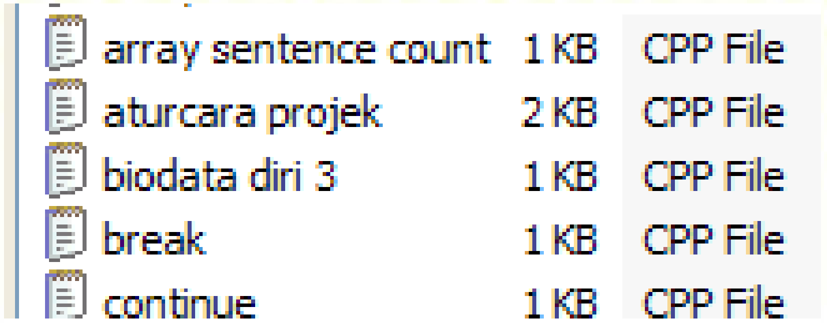
```
#include <iostream.h>
#include <conio.h>
void main ()
{
char nombor[] = {'a','b','c','d',} ;
cout<<nombor[2] ;
getch();
}
```

source code



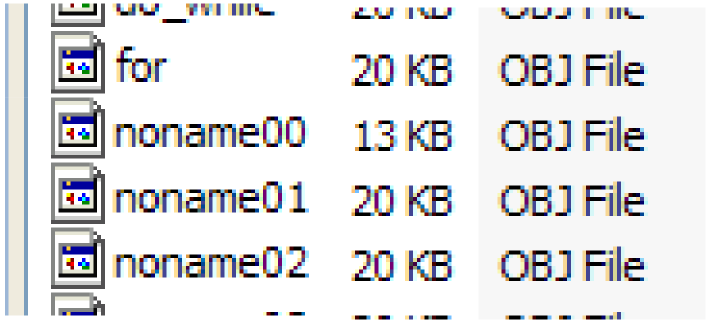
array sentence count	1 KB	CPP File
aturcara projek	2 KB	CPP File
biodata diri 3	1 KB	CPP File
break	1 KB	CPP File
continue	1 KB	CPP File

*.cpp file



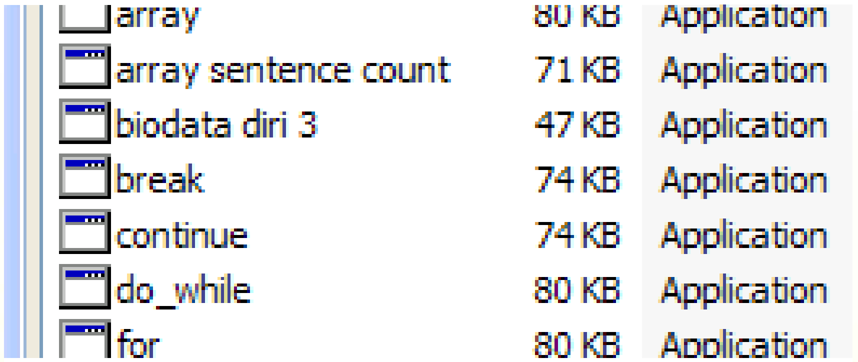
array sentence count	1 KB	CPP File
aturcara projek	2 KB	CPP File
biodata diri 3	1 KB	CPP File
break	1 KB	CPP File
continue	1 KB	CPP File

*.cpp file



do_while	20 KB	OBJ File
for	20 KB	OBJ File
noname00	13 KB	OBJ File
noname01	20 KB	OBJ File
noname02	20 KB	OBJ File

*.obj file



array	80 KB	Application
array sentence count	71 KB	Application
biodata diri 3	47 KB	Application
break	74 KB	Application
continue	74 KB	Application
do_while	80 KB	Application
for	80 KB	Application

*.exe file

Environment of C++

- There are several important terms that has certain functions in the C++ language environment, among them are:
  - Text editor
  - Compiler
  - Debugger
  - Linker
  - Make

Integrated Development Environment (IDE)

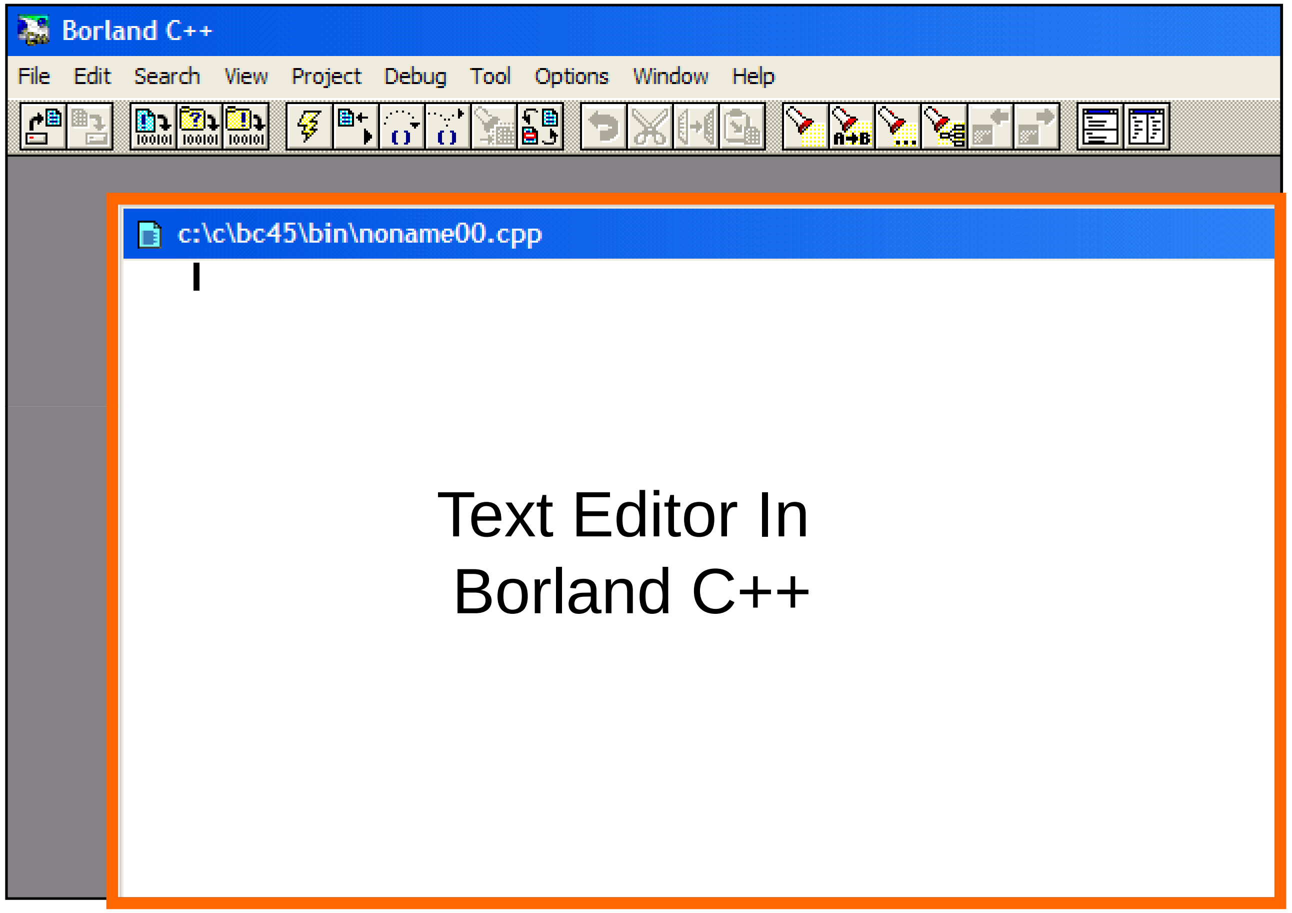
- **Text editor**
  - Allows writing and editing activities of C++ programming codes
  - Notepad (simple editor), emacs (UNIX), pico
- **Compiler**
  - Converting the source code to object code that is understandable by the CPU
  - DOS/Windows
    - Borland C/C++
    - Microsoft Visual C/C++
  - UNIX - GNU C/C++ compiler

- **Linker**
  - Converting the object code into .exe files.
  - Merging all the necessary parts (e.g: library files) by the program to produce the final codes in the form of .exe to be executed/run

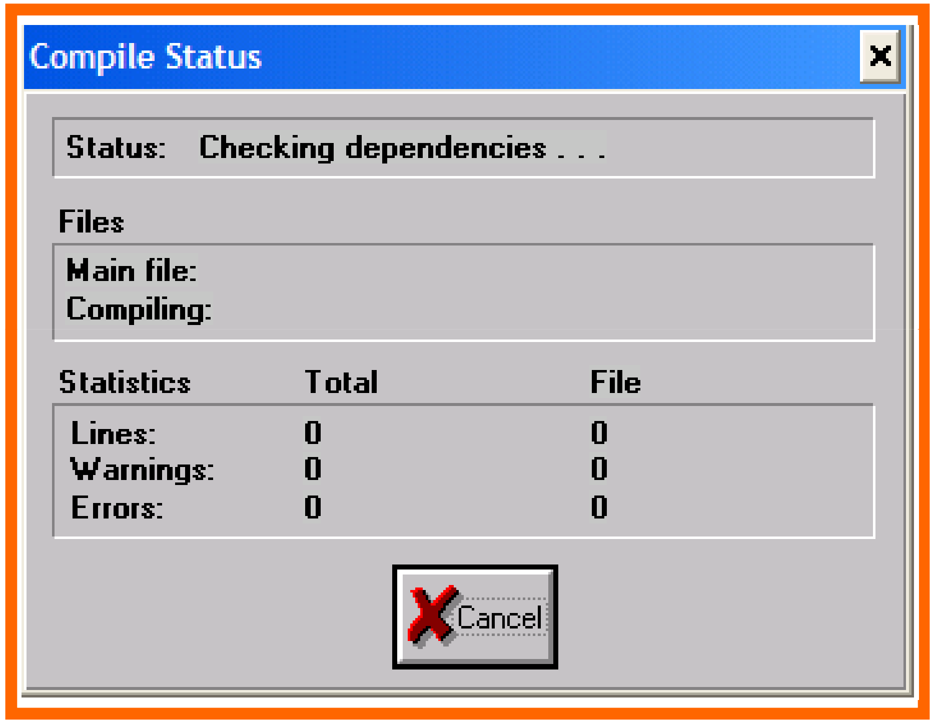
- **Debugger**
  - An application used to analyze the program
  - Identifies errors and mistakes in the program

- **Make**
  - A utility program that is used in C/C++ project development

- **Integrated Development Environment (IDE)**
  - Integrates editing activity, compiling, debugging and testing in a single environment
  - Simplifying programming project management like Turbo C++ / Borland C++


Borland C++
File Edit Search View Project Debug Tool Options Window Help
c:\c\bc45\bin\noname00.cpp

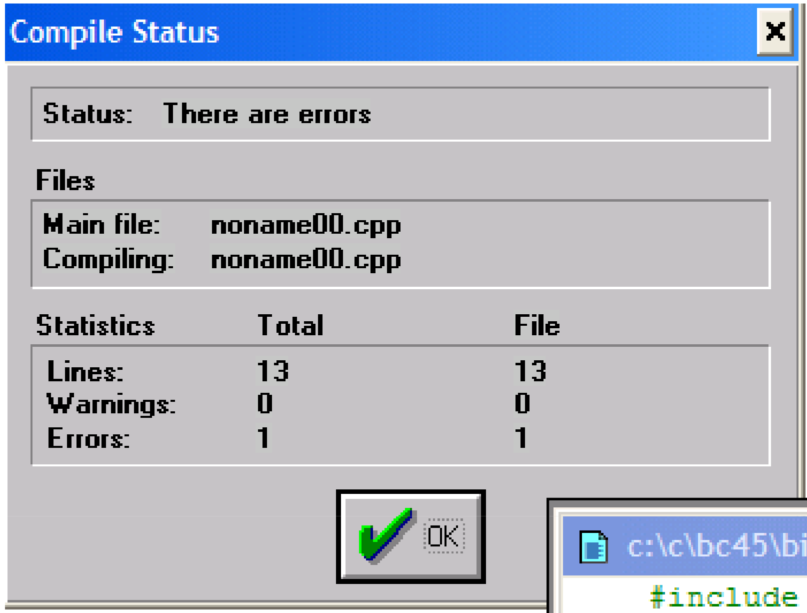
Text Editor In
Borland C++



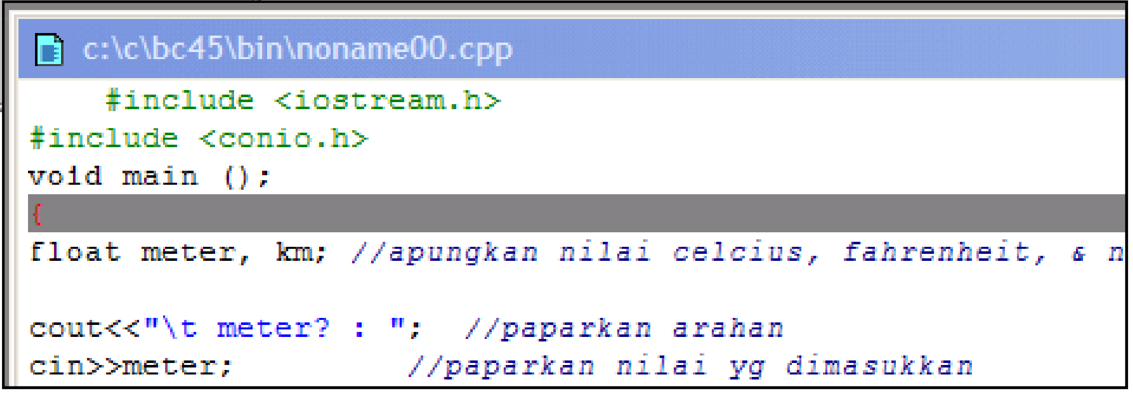
```
Compile Status

Status:   Checking dependencies . . .

Files
Main file:
Compiling:

Statistics          Total          File
Lines:              0              0
Warnings:           0              0
Errors:             0              0
```

Compiler in C++

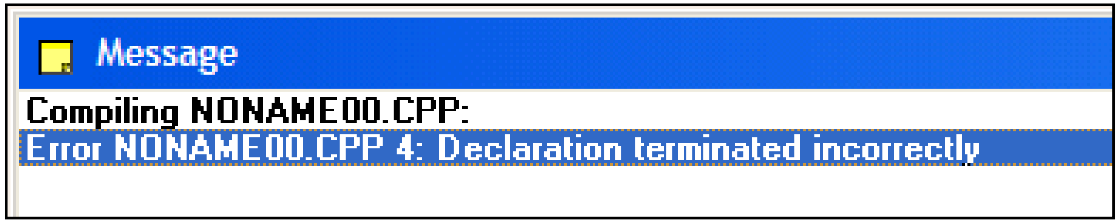
Debugger in C++



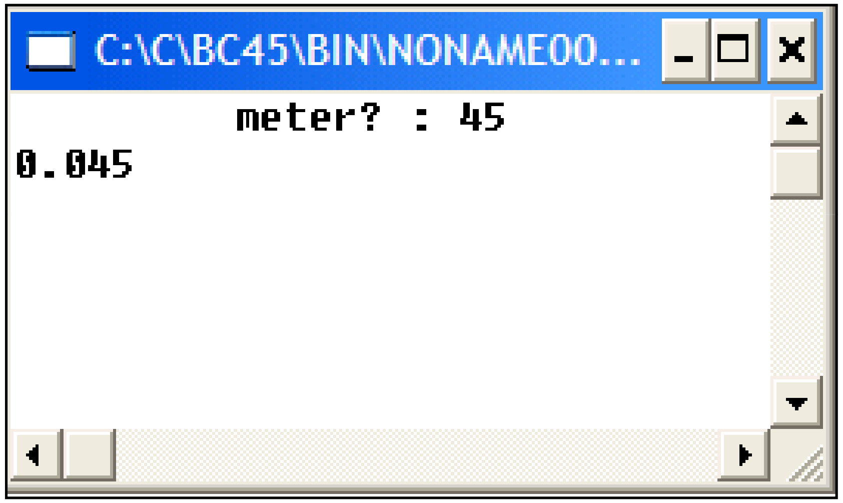
```
Compile Status

Status:   There are errors

Files
Main file:    noname00.cpp
Compiling:    noname00.cpp

Statistics       Total          File
Lines:           13             13
Warnings:        0              0
Errors:          1              1
```



```
c:\c\bc45\bin\noname00.cpp
    #include <iostream.h>
#include <conio.h>
void main ();
{
float meter, km; //apungkan nilai celcius, fahrenheit, & n

cout<<"\t meter? : ";   //paparkan arahan
cin>>meter;          //paparkan nilai yg dimasukkan
```

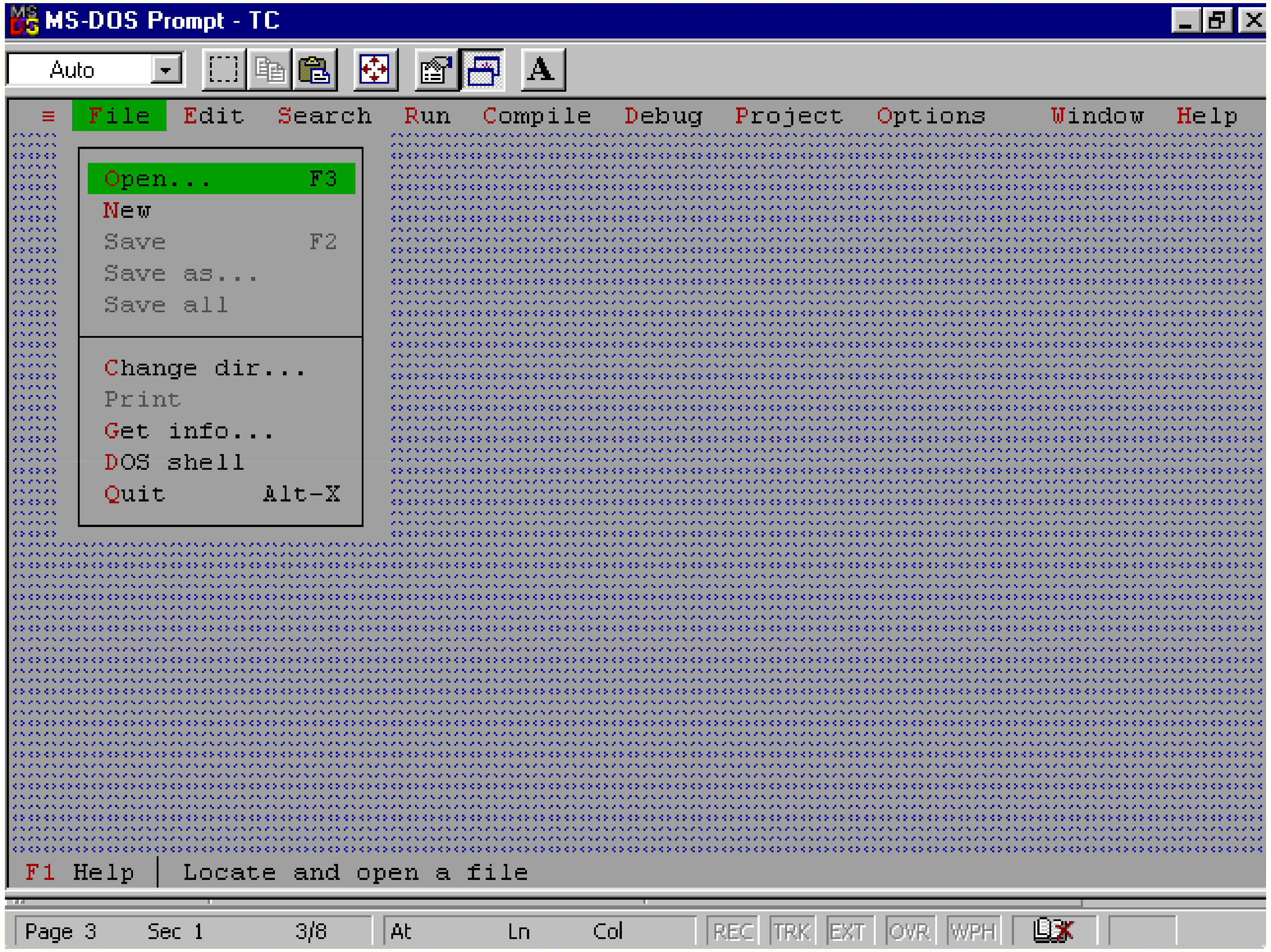


```
Message
Compiling NONAME00.CPP:
Error NONAME00.CPP 4: Declaration terminated incorrectly
```



```
          meter? : 45
0.045
```

exe file - Borland C++



Locate and open a file

An IDE of Turbo C++

- A C++ program will have the basic structure as follows:
  - Comments - //
  - Preprocessor directives - #include <conio.h>
  - Main function / void main ()
  - Variable declaration / int no1, no2;
  - C++ statement / cout<<no1;
  - Return statement / return no1;

- Comments
  - Writable in any part of the program
  - It will not result in any action by the computer (compilers do not process comments)
  - Used to make the program easier to be read and understand. Also used to explain any part of the program as well as documentation.
  - Written in between **/*** and ***/** or after // as you can observe below:
  - /*…*/ mark
    - ex: /* My first programming */
  - // mark
    - ex: // My first programming

- Preprocessor directives
  - Starts with #
  - Used to include *header file/s*
  - The form of preprocessor directives is:
    - *#include<header file>*
  - The #include<iostream.h> directive is a direction to include the header file for **stream input-output** that contains the definition for **cout** and **cin**

    iostrem – Input Output Stream

    Cin>> Cout<<

- Main () function
  - A block code that runs a task
  - Every C++ program must have one main() function
  - Consists of head and body
    - The head contains preprocessor definitions and instructions
    - Also contains the basic preparations for the related functions
    - The body part contains programming codes for the main() function
    - Decides what actually the function does here

- The form of a main() function for a C++ program is as follows:
  - Main() function type

    { C++ statement…; }

  Ex:

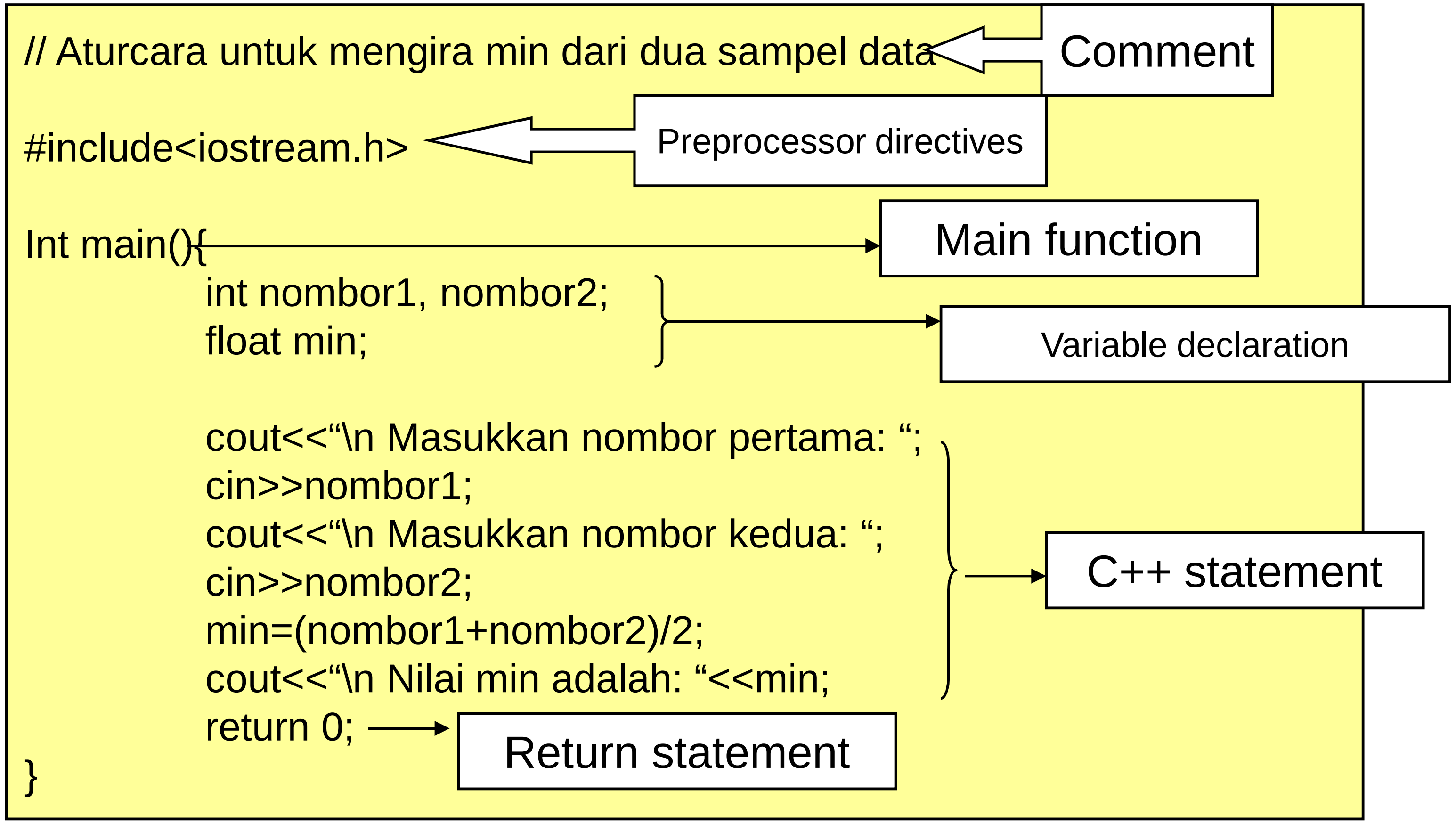
```
#include <iostream.h>
#include <conio.h>
void main ()
{
cout<<" arahan ";  //paparkan arahan
getch();
}
```

Return statement

- Written at the end of a program where it will divert the control from the program to the OS
- Return 0, means that the program could be executed without error
- Functions that uses *void,* there will be no value returned to the OS
- Eg :

```
#include <iostream.h>
main ()
{
cout<<" Hai ";
return(0);
}
```

Example



```
// Aturcara untuk mengira min dari dua sampel data

#include<iostream.h>

Int main(){
            int nombor1, nombor2;
            float min;

            cout<<"\n Masukkan nombor pertama: ";
            cin>>nombor1;
            cout<<"\n Masukkan nombor kedua: ";
            cin>>nombor2;
            min=(nombor1+nombor2)/2;
            cout<<"\n Nilai min adalah: "<<min;
            return 0;
}
```

C++ *Statements*

- Instructs the computer to take action
- There are two types of C++ statement
  - Phrase Statement
    - Represents data such as numbers or characters or even an entity like combination of variables
    - Ex:

      Pay_sum = total_hours * pay_rate
  - Control Statement
    - Consists of linear, selection and looping statements

C++ statement ending

- Every C++ statement must be ended with a semicolon (;)
- The semicolon acts as an ending
- Without the semicolon, the compiler will inform that there is an error in the program/compiling process
- eg : cout<<“Hello”
- A preprocessor directive does not need an ending (;)
  eg : #include <iostream.h>

C++ statement ending

```
#include <iostream.h>
#include <conio.h>
void main ()
{
cout<<" arahan “;

cout<<" arahan 2“;

cout<<" arahan 3“;
getch();
}
```

Semicolon

Variable And Constant In C++ Programming Language

Variable

- Define & declare by user (eg : int numb, char name[2])
- Uniquely on the scope
- Never start with number
- Used underscore (_) for spacing
- never use space between char
- Never use special symbol (eg : %$|&^><:}*/^%)
- Case sensitive

Variable

- int no1, no2 ;
- char name_1[5];
- *int x,X,x2;*
- *cin>>x;*
- *cin>>X;*
- *x2=x+X;*
- *cout<<x2;*

Constants

- *Constants* are expressions with a fixed value.
- You can define your constants that you use very often by using the #define preprocessor directive. Its format is:

  #define identifier value
  For example:

  #define PI 3.14159
- *#define NEWLINE '\n'*

  This defines two new constants: *PI* and *NEWLINE*. Once they are defined, you can use them in the rest of the code