

IISEE lecture for group training

Fortran programming for beginner seismologists

Lesson 1

Lecturer

Tatsuhiko Hara

Reference

Introduction to FORTRAN90/95 by S. J. Chapman (New York: McGraw-Hill, 1998)

Why Fortran?

- Many aspects common among computer programming languages
- Many seismological software have been developed using Fortran.
- Programming is effective to improve your understanding (it is impossible to write a code without proper understanding).

Starting editor

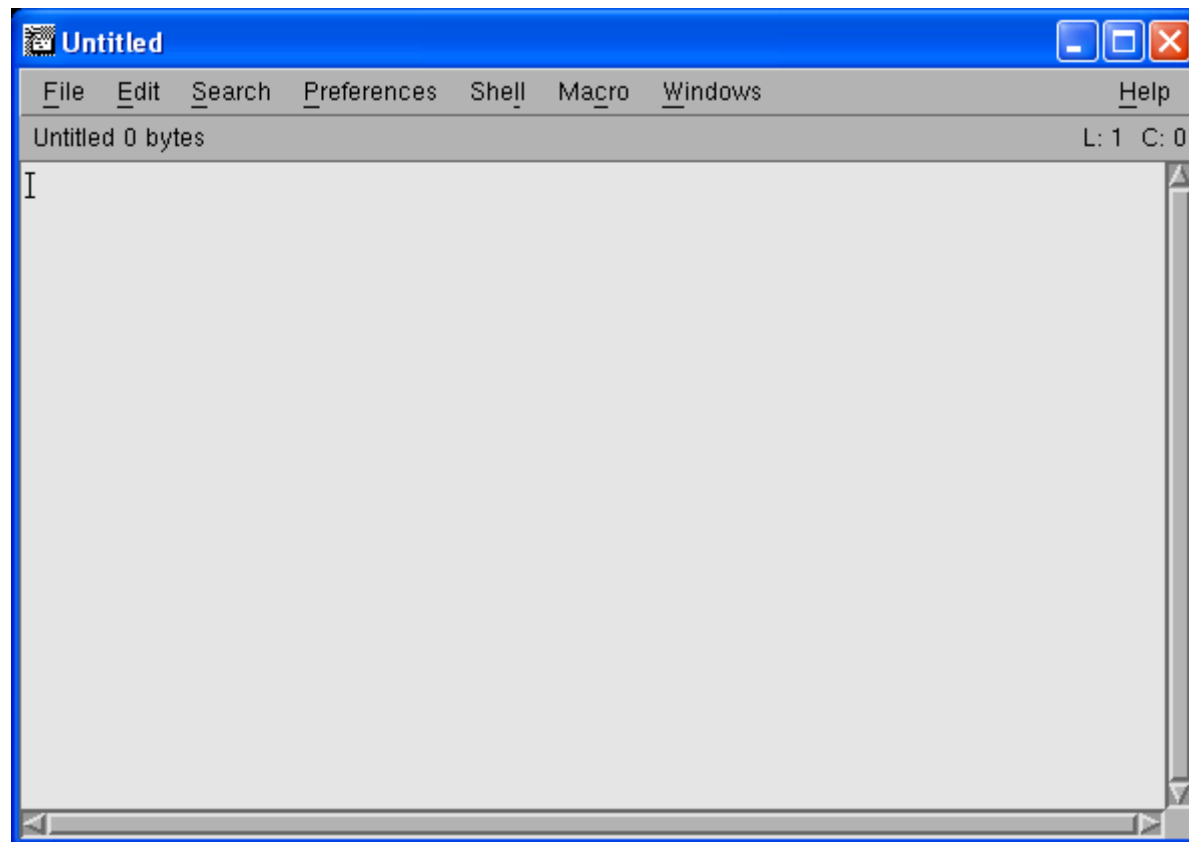
- We use “nedit” to edit programs.
- Type as:

```
$ nedit &
```

then you will get the window shown in the next slide.

Note: When you put “&” after the command, that command is executed as a background job.

nedit

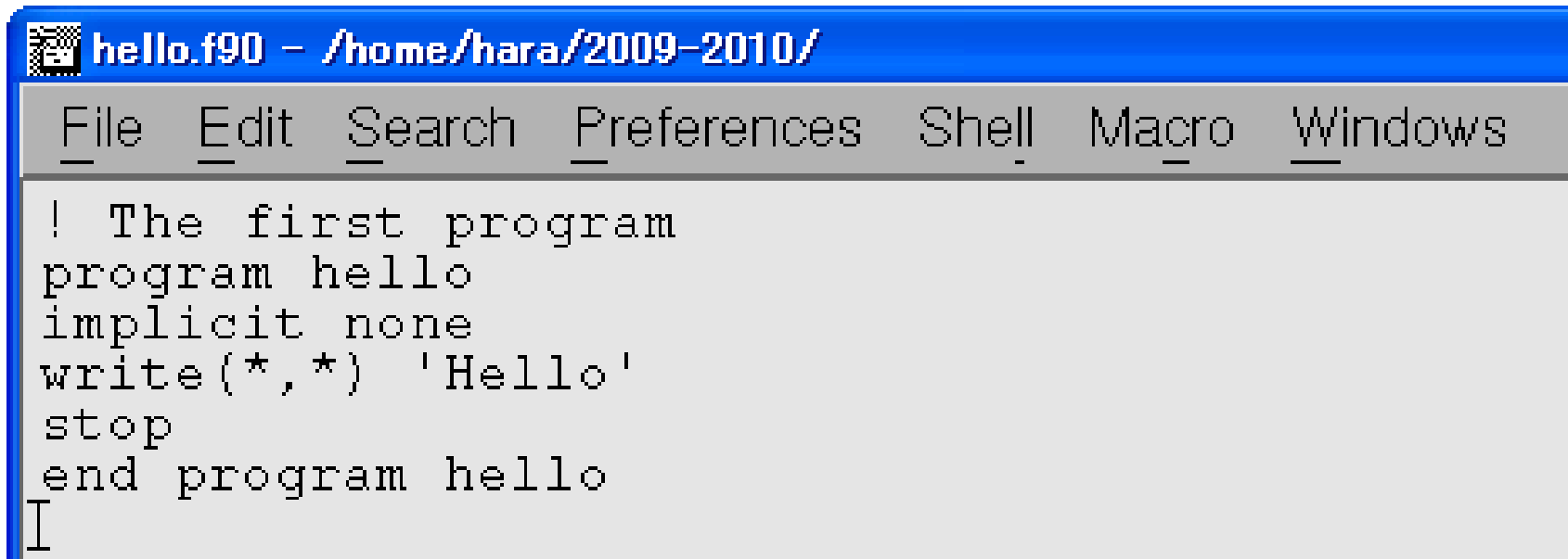


The first program

- Write the following in the nedit window

```
! The first program
program hello
implicit none
write(*,*) 'Hello'
stop
end program hello
```

In the actual editor



The image shows a screenshot of a text editor window. The title bar at the top is blue and contains the text "hello.f90 - /home/hara/2009-2010/". Below the title bar is a menu bar with the following items: File, Edit, Search, Preferences, Shell, Macro, and Windows. The main editing area is light gray and contains the following Fortran code:

```
! The first program
program hello
implicit none
write(*,*) 'Hello'
stop
end program hello
I
```

Save the file

- Save the program as `hello.f90`:
 - Click “File” → “Save As” in the editor
 - Then write the name of the file
 - Click “OK”

EXERCISE 1-1

- a) Use a `ls` command in the xterm to confirm that the new file is created.
- b) Type as “`cat hello.f90`” to see the content.

Compile and Execution

- Type the following command to compile the program:

```
$ gfortran hello.f90
```

- Use a *ls* command to confirm that “a.exe” is created.
- Then type as:

```
$ ./a.exe
```

Note: The last command is specified by the relative path. “.” denotes the current directory.

The structure of a Fortran program

```
! The first program
program hello
implicit none
write(*,*) 'Hello'
stop
end program hello
```

} Declaration section

} Execution section

} Termination section

The Fortran character set

Uppercase letters of the alphabet	A through Z	26
Lowercase letters of the alphabet	a through z	26
Digits	0 through 9	10
Underscore character	_	1
Arithmetic symbols	+ - * / **	5
Miscellaneous symbols	() . = , ' \$: ! " % & ; < > ? \$ and blank	18

Fortran statements are case insensitive.

Program name

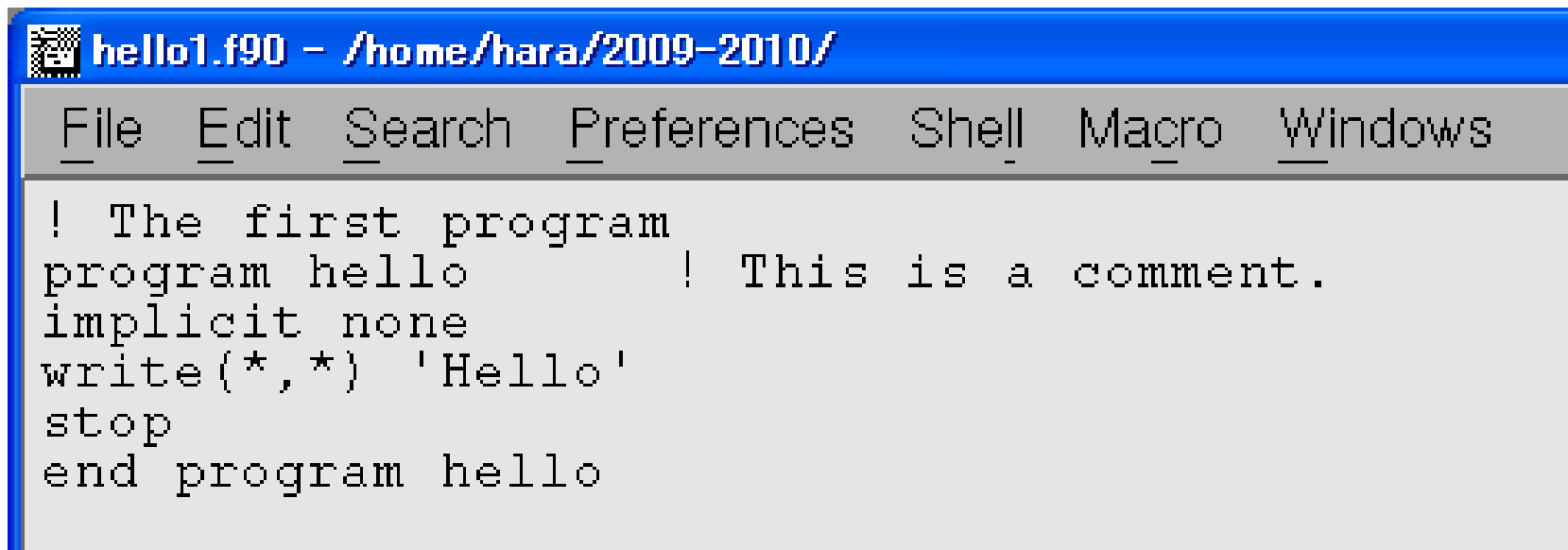
- Fortran program names may be up to 31 characters long.
- Any combination of alphabet characters, digits, and the underscore (_) character starting from an alphabet character can be used.

Comment lines

- When you put “!” in a program, the part after “!” is interpreted as comments.

EXERCISE 1-2

Modify “hello1.f90” as below. Then, compile and execute the modified program.



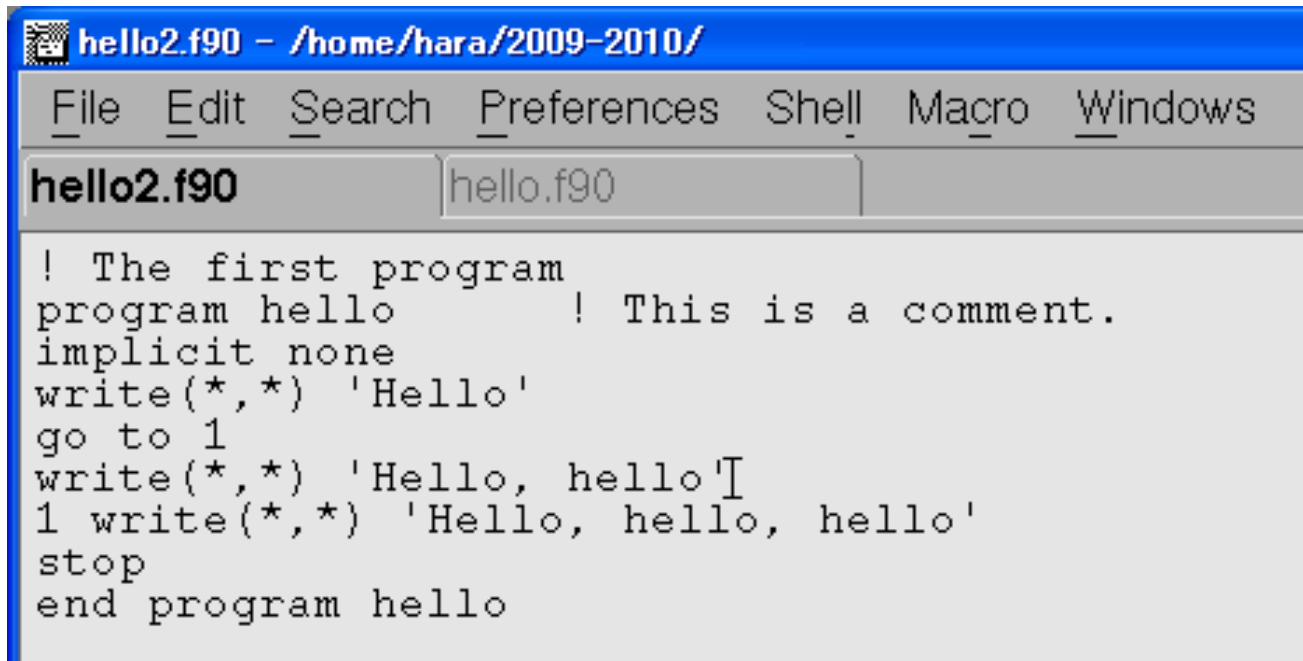
```
hello1.f90 - /home/hara/2009-2010/  
File Edit Search Preferences Shell Macro Windows  
! The first program  
program hello      ! This is a comment.  
implicit none  
write(*,*) 'Hello'  
stop  
end program hello
```

Statement labels

- A number at the beginning of a line is called statement label. This is the “name” of a Fortran statement, and can be referred to a statement from other parts of the program.

EXERCISE 1-3

Modify “hello1.f90” as below and save it as “hello2.f90”. Then, compile and execute the modified program.



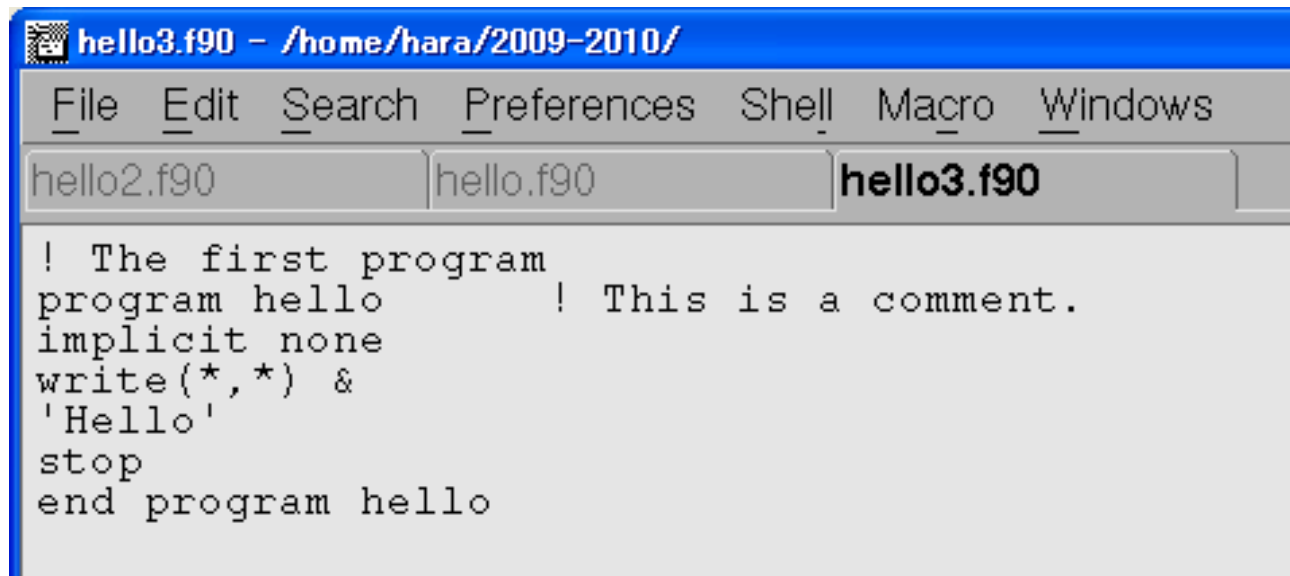
```
hello2.f90 - /home/hara/2009-2010/  
File Edit Search Preferences Shell Macro Windows  
hello2.f90 hello.f90  
! The first program  
program hello      ! This is a comment.  
implicit none  
write(*,*) 'Hello'  
go to 1  
write(*,*) 'Hello, hello'  
1 write(*,*) 'Hello, hello, hello'  
stop  
end program hello
```

Continuation of lines

- When you end a line with an ampersand (&) character, you can continue that statement on the next line.

Exercise 1-4

Modify “hello1.f90” as below and save as “hello3.f90”. Then, compile and execute the modified program.



The screenshot shows a text editor window titled "hello3.f90 - /home/hara/2009-2010/". The window has a menu bar with "File", "Edit", "Search", "Preferences", "Shell", "Macro", and "Windows". Below the menu bar are three tabs: "hello2.f90", "hello.f90", and "hello3.f90". The main text area contains the following Fortran code:

```
! The first program
program hello      ! This is a comment.
implicit none
write(*,*) &
'Hello'
stop
end program hello
```

Where can we put statements?

- FORTRAN statements can be entered anywhere on a line (free-source form).
- Each line can be up to 132 characters.

EXERCISE 1-5

Enter statements beyond the 132nd column, and compile that program.

WRITE statement

- The following statement
`write(*,*) expression [expression, etc.]`
writes the values of one or more expressions to the specified output device.
- In the above statement, the standard output device (i.e., the screen) is specified by the first “*”.
- The standard format is specified by the second “*”.