Introduction to FORTRAN and C

Suzanne Parete-Koon

OLCF

Managed by UT-Battelle
for the Department of Energy

OAK RIDGE National Laboratory

C and Fortran basics

- Variables
- Selection
- Loops
- Functions
- When this hour is over you should be able to write a “hello world” in C and Fortran

Step 0 : The text editor

- You will be writing this code with a text editor. I am going to use vi and give you a quick tutorial.
- To create an open a file with vi:
- Vi filename.f90
- To write in vi, hit i, to stop writing hit esc.
- To save hit esc :w
- To quit hit esc :q
- OK your turn: Open a file calle hello.f9
- Vi hello.f90

Fortran Hello World

vi hello.f90

Hit “i” to turn on vi’s text insert.

Enter the following:

```
Program hello
      write(*,*) “Hello World”
End
```

Hit esc to turn off vi’s insert.

Hit :wq to write (save the file) and quit vi.

To compile: `ftn hello.f90`

To run: `./a.out`

Fortran Hello World

```
Program hello
      write(*,*) “Hello World”
End
```

The basic Fortran program block begins with “Program” and ends with “end”.

write(*,*) – means write in the default format, to the screen.

Fortran Hello World

```
nk8 — ssh — 80×24
[titan-ext5] [03:16:20] [~/crashcourse/fortran] $ftn hello.f90
[titan-ext5] [03:17:46] [~/crashcourse/fortran] $./a.out
 Hello World!
[titan-ext5] [03:22:39] [~/crashcourse/fortran] $
```

Fortran Variables

- FORTRAN supports six different data types, Each type you use is declared at the top of your program.
- Integer !32 bits, 10
- Real !32, 64 bits, 10.123456
- Double precision !64 bits, 10.123456789121112
- Character, “hello”
- Complex , (5.229,-4.78)
- Logical , .true.

Arithmetic Operations

- + Addition z= y+x
- - Subtraction y= z-x
- * multiplication z=y*x
- / Division y=z/x
- ** Exponentiation three_squared= 3**2

Fortran Helloworld+

hello+.f90

```
program hello
      implicit none ! All variables declared
      integer x     ! Declare x as an integer
      real y,z      ! Declare y as a real
      character*5 name    ! Length 5
           x=10
           y=3.14159
           z=y*2
           name="hello"
           write(*,*)x,y,z,name !use commas
                                !for multiple
                                !variables
end
```

Fortran Hello World+

nk8 — ssh — 80×24

```
[titan-ext2][04:10:24][~/crashcourse/fortran]$ftn hello+.f90
[titan-ext2][04:11:21][~/crashcourse/fortran]$./a.out
           10    3.141590118408203          6.283180236816406      hello
[titan-ext2][04:11:28][~/crashcourse/fortran]$
```

Fortran Your Turn: Dog years

- If a dog ages 7 dog years for each human year, write a program that coverts human years to dog years and writes the result to the screen. Assume the dog is 9 human years old.

Fortran Your Turn: Dog years

- What will you need to do first for FORTRAN ?
- What variables will you need to declare?
- Fido is 9 human years.
- Fido ages 7 dogyears for every human year.
- How will you write your answer?
- What do you need to do last for FORTRAN?

Fortran Your Turn: Dog_years.f90

```
program dogyears
 ! If a dog ages 7 dog years for each human year,
 ! write a program that coverts human years to dog
 ! and writes the result.

 implicit none
 !declare two integers, h and d, to hold human/dog

 integer d,h
 ! set the dog's age to 9 human years
   h=9
 ! convert human years to dog years
   d=7*h

write(*,*) "In dog years, Fido is ",d,"."

end
~
```

Fortran Loops

Do, i= min, max

do something that depends on i

enddo

Example: 10 years of dog age conversions:

```
Do h=1,10

   d=7*h

   write(*,*) “Fido is”, d, “dog
years old.”

enddo
```

Fortran Selection

- If condition then do something, else do something else.
- Example more accurate dog years

```
if(h < 3)then
     d=10*h
  else
     d=20+7*(h-2)
endif
```

C Hello World!

```
#include <stdio.h>

void main(void)

{

    printf("Hello World\n");

}
```

To compile: `cc hello.c`

To run : `./a.out`

Variable Declaration

- int I;
- char c;
- double dbl;
- float f;
- int i=0;
- const pi=3.14159265;

Hello World +

hello+.c

```
#include <stdio.h>

void main(void)

{

    int x;

     float y;

     x=10;

     y=3.14159

     printf(“Hello World\n”);

     printf(“x=%d y=%f”,x,y);

}
```

Loops C

- **For expressions**

```
for (h=0; h<10; h++)
  {
   dog=10*h;
   printf("Fido is %d h_years; %d d_years \n",h,dog);
   }
```

C Selection

- **if condition, then something, else something else.**

```
if (h < 3)
{
   dog=h*10;
}
else
{
 dog=(h-2)*7 +20;
}
```

C your turn: hello Bobby

```
int main(int argc, char** argv)
```

- **argc and argv allow you to pass arguments in to main.**
- **argc- argument count**
- **argv – the list of arguments**

C your turn: hello Bobby

hello_bob.c

```
#include <stdio.h>

int main(int argc, char** argv)

{

    printf("Hello %s \n", argv[1]);

    return (0);

}
```

To compile: `cc hello_bob.c`

To run: `./a.out Bobby`

C your turn: hello Bobby

`Hello_bob.c`

C Pointers on Pointers

- **A pointer stores the address of a variable**
- **Pointer_demo.c**

```
int a;

int *a_pointer;  //a pointer

a = 5;

a_pointer = &a; // "&" means store the\

                //  address of "a”

printf("the address of a = %p, the value of a =
%d\n", a_pointer, *a_pointer);
```

C or Fortran Your Turn

- Modify dog_years to loop over 0 to 9 human years and print out the conversion to dog years for each i.
- If you are really into this, put in a condition to multiply the dog's age by 10 for the first 2 years and then, by 7 for every year after.

Questions?

C your turn: hello Bobby

```
#include <stdio.h>

int main(int argc, char** argv)

{

  printf("Hello %s %s\n", argv[1],argv[2]);

    return (0);

}
```

To compile: cc hello++.c

To run: ./a.out Bobby Suzanne