CS19003: Programming and Data Structures Laboratory

Aritra Hazra,
CSE Dept., IIT Kharagpur

http://cse.iitkgp.ac.in/~aritrah/course/lab/PDS/Spring2021/

06-Apr-2021

Hello World : Our First C Program

- Source Code: file `hello.c` contains the following lines:

```
/***********************************
* Name :  YOUR FULL-NAME
* Roll No :  20CS100XY / 20CS300XY
* Section :  10
* Assignment No :  0
* Description :  First C program
* Date :  30-Mar-2021
***********************************/

#include <stdio.h>
int main()
{
    /* check the indentation */
    printf("hello world\n");
    return 0;
}
```

- Execution Output: shall print as output the string hello world.

Hello World : Program Explanation

- `/*...*/` : comment line not executed with program
- `#include <stdio.h>` : tells the compiler to include information about the standard i/o library
- `int` : a *data type* of C indicating integers
- `main()` : a special *function*. Every C program begins execution from the first line in main
- `printf()` : a *library function* that prints output
- `"hello world"` : character string to be printed
- `\n` : *newline character*
- `return` : value returned from the function `main()`

Getting input from keyboard

- `scanf("%d", &n)` : reads one integer from the keyboard and stores it in variable n
- `scanf("%d%c", &n, &c)` : reads one integer from the keyboard and puts it in n, and reads one character from keyboard and puts it in c
- scanf("put the format string here", list of variables each prefixed with &);

Some possible formats

- %d : to read/write int (integer)
- %f : to read/write float (floating point numbers)
- %lf : to read/write double (large floating point numbers)
- %c : to read/write char (a single character)

printf statement

- Specify a format to print the data
- Example:
  printf("The two numbers read are %d and %d\n", a, b)
- can print any string, including blanks, to make your output look nicer
- Unlike scanf, no & to be given before variable name

Examples

```
int main()
{
 int n;
 scanf("%d", &n);
 printf("The no. read is %d \n", n);
 return 0;
}
```

```
int main()
{
 char c;
 scanf("%c", &c);
 printf("Read one character %c \n", c);
 return 0;
}
```

Examples

```
int main()
{
 float n;
 printf("Enter a floating point number: ");
 scanf("%f", &n);
 printf("The number %.2f is read \n", n);
 return 0;
}
```

If you input 23 from keyboard, this will print
"The number 23.00 is read"

Data Types

Programs operate on 'data' which is stored in 'variables' which are classified into 'data types' depending on memory storage and nature of data.

- `int` : basic signed integer type.
- `char` : can hold a single character. A character is what you get when you press a key on a keyboard.
- `float` : storing real numbers with limited precision.
- `double` : storing real numbers with finer precision.

Variable declaration, initialization, operation


```
int main ( )
{
 int first, second, third;
 first = 1;
 second = 2;
 second = second + first;
 return 0;
}
```

C expressions

An expression is a program statement containing variables and constants and operators in the right order so that it can be 'evaluated' by the computer. The result may be assigned to another variable

- Arithmetic expression : operators : "- ,+ ,* ,/ "
  - Ex: $z = x + y * 4$; n = 2*(3+5) - 4/2;
- Relational expression : operators :

  `>, <, ==, >=, <=, !=`
  - `Ex: a >= b;`
- Boolean expression : combining relational expressions using Boolean connectives &&, ||
  - `Ex: (a > b && a > c)`

Example program with arithmetic expression

```
#include <stdio.h>

int main()
{
  int x1, x2, x3, x4, z;

  scanf("%d%d%d%d", &x1, &x2, &x3, &x4);
  z = (x1 + x2 + x3 + x4)/4
  printf("The average is %d \n", z);
  return 0;
}
```

Example program with arithmetic expression

```
#include <stdio.h>
#include <math.h>
int main()
{
  int x1, y1, x2, y2;
  double len;
  scanf("%d%d%d%d", &x1, &y1, &x2, &y2);
  len = sqrt( (pow(x1-x2, 2)) + (pow(y1-y2, 2)) );
  printf("The line length is %f \n", len);
  return 0;
}
```

- A whole bunch of mathematical functions are already defined that can be used. Ex: sqrt, pow, cos, sin, etc
- Note that the value returned by sqrt is stored in variable of type double (this is true for most math functions)
- For functionalities in math.h: `www.cplusplus.com/reference/cmath/`

Types of Data in Expressions

- When C performs an operation, it makes a guess as to the type of the result that is to be produced.
- Essentially, if the two operands are integer, it says that the result should be integer, if the two are floating point, it says that the result should be floating point.
- $1/2 \neq 1/2.0$
- C evaluates $1/2$ as 0 and $1/2.0$ as 0.5

Type casting

- We can force C to regard a value as being of a certain type by the use of casting.
- You cast a value by putting the type you want to see there in brackets before it.

```
#include <stdio.h>
int main ( )
{
    int i = 3, j = 2 ; float fraction ;
    fraction = (float) i / (float) j ;
    printf ( "fraction : %f\n", fraction ) ;
    return 0;
}
```

The (float) part of the above tells the compiler to regard the values in the integer variables as floating point ones, so that we get 1.5 printed out rather than 1.0.

Thank You