```
CSE1062 | CCS1063 ‘Practicals’ {

    [Fundamentals of Computer Programming]

        < Tutorial Session 01 - Intro >

}
```

INTRODUCTION TO C {

01 Developed early 1970 by Dennis Ritchie at Bell Laboratories

02 Mother of ALL Languages

< C++ (also known as C with classes), C#, Python, Java, JavaScript, Perl, PHP, Verilog, D, Limbo and C shell of Unix etc. >

< Python uses C for creating standard libraries, whereas the syntaxes and control structures of languages like C++, PHP and Perl are based on C. >

}

What is C {

- General purpose, high level, structured programming language.
- C mostly used for writing operating systems and other system programming.
- It can be used for developing applications.

}

What is C …

C is a case sensitive language.

Each statement should end with a semicolon ;

There can be more than one statements in a line.

```
int x=5; char a = 'A';
```

Create your C program

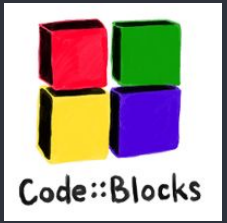
IDEs (Integrated Development Environments)

- ➔ Code::Blocks
- ➔ Dev-C++

We will use

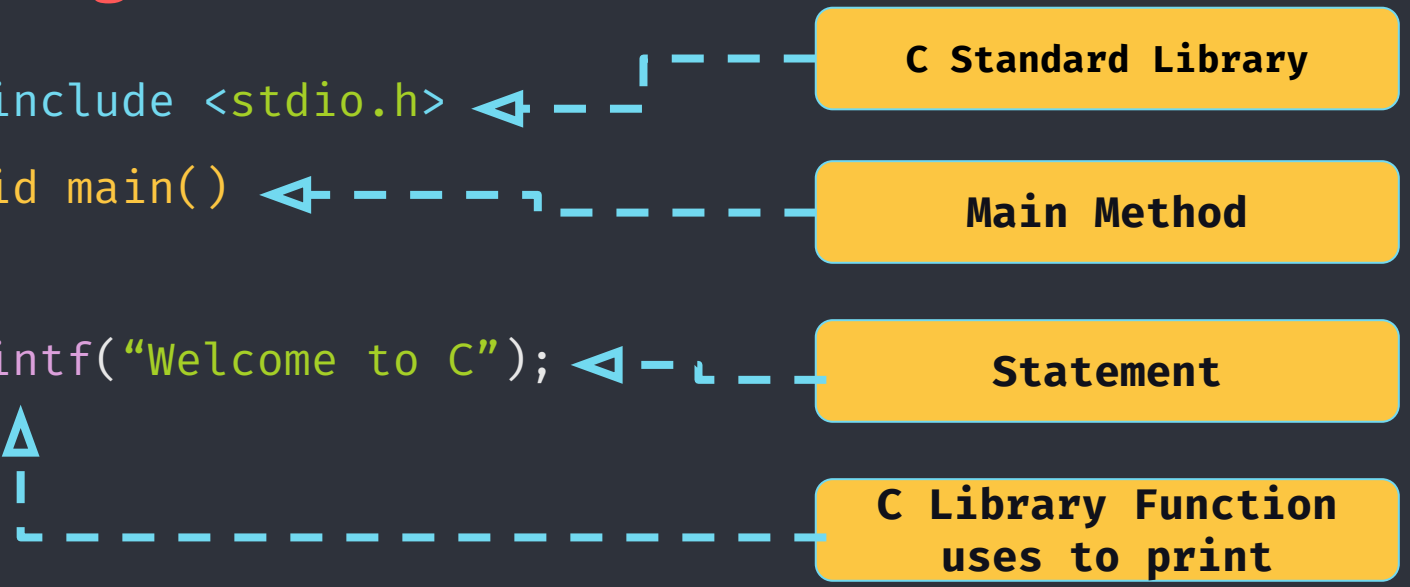
GNU Compiler Collection (GCC)

Comes up with Ubuntu



A Simple C Program: Display a message on screen

```
1
2
3
4
5   # include <stdio.h>
6
7   void main()
8   {
9
10  printf("Welcome to C");
11  }
12
13
14
```



A Simple C Program: Display a message on screen

#include <stdio.h>

- Large number of functions available in C, which can be used directly

- Those are called as library functions, which are written as C programs and saved in the folder called **“include”**

- <stdio.h> allows standard input/output operations

https://www.geeksforgeeks.org/c-library-functions/

A Simple C Program: Display a message on screen

```
void main()
```

- C programs contain one or more functions, exactly one of which must be main
- Parenthesis used to indicate a function ()
- Braces ({ and }) indicate a block
- The bodies of all functions must be contained in braces
- There are a equal number of opening and closing braces

A Simple C Program: Display a message on screen

```
printf ( "Welcome to C!" );
```

- Instructs computer to perform an action
  - Specifically, prints the string of characters within quotes (“ ”)
- Entire line called a statement
  - All statements must end with a semicolon (;)

Comment Statements

Comments may appear anywhere in the program

➔ Single line comments:

- Denoted by // anything written after this in the same line will be comments and will be ignored by the compiler

➔ Multiple line comments:

- Denoted by /* */ anything written between /* and */ will be ignored by the compiler.

Escape Characters

\a	**Alert (bell, alarm)**	\\	**Backslash**
\b	**Backspace**	\'	**Single Quote**
\n	**New Line**	\"	**Double Quote**
\t	**Horizontal Tab**	\?	**Question Mark**
\v	**Vertical Tab**	\000	**Octal Number**
\f	**Form Feed**	\xhh	**Hexadecimal Number**

Example 01 :

```c
#include <stdio.h>
void main() {
printf("This is a line of text.\n"); // Newline character '\n'
printf("This\tis\ta\ttabbed\tline.\n"); // Tab character '\t'
printf("This is a backslash: \\.\n"); // Backslash character '\\'
printf("This is a double quote: \"Hello!\"\n"); // Double quote
character '\"'
}
```

Output:

```
This is a line of text.
This    is    a    tabbed    line.
This is a backslash: \.
This is a double quote: "Hello!"
```

The keyword set of C language

As C is a case sensitive language, all keywords must be written in lowercase.

auto, else, register, union, break, enum, return,

unsigned, case, extern, short, void, char, float,

signed, volatile,const,for,size of, while,continue,

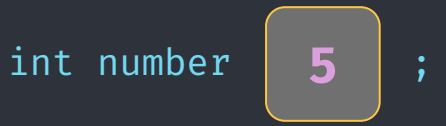
goto, static, default, if, struct, do, int, switch,

double, long, typedef

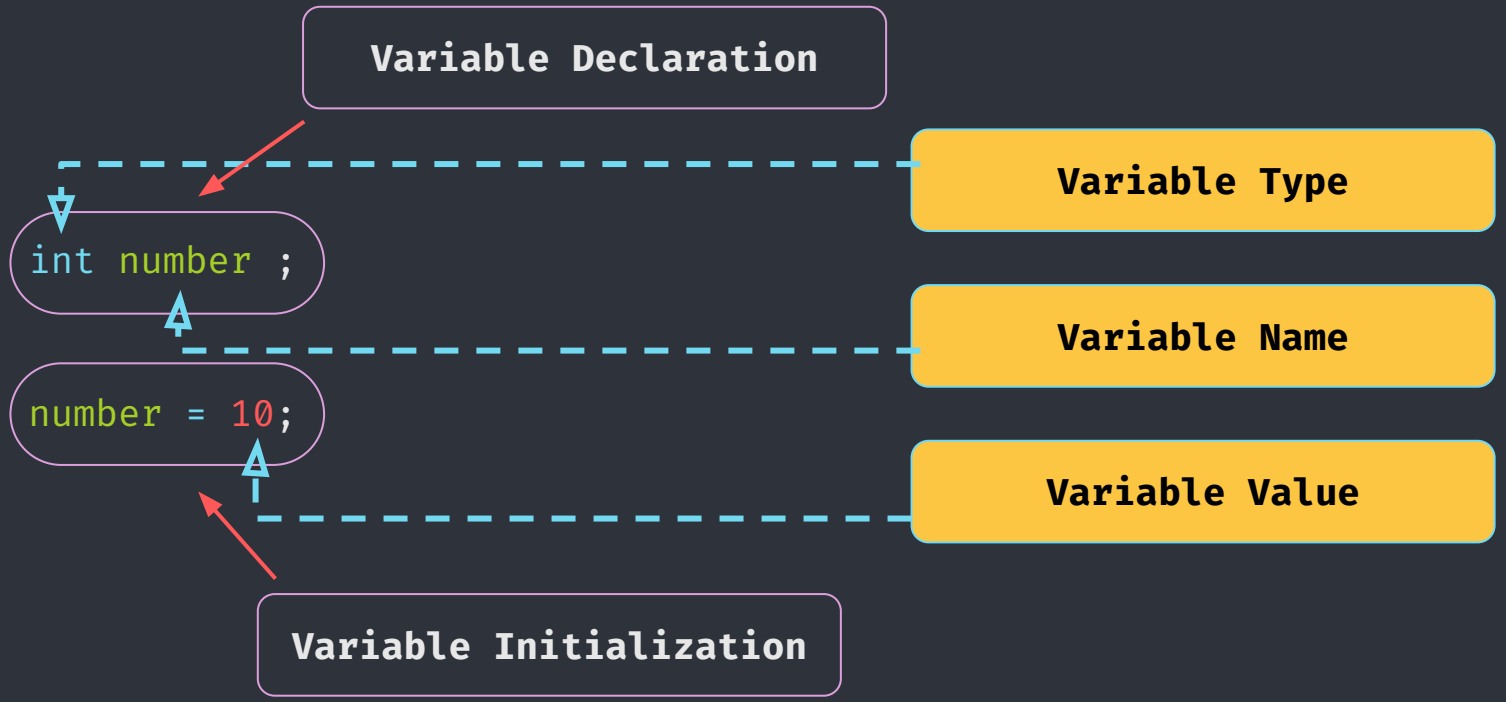
You can't use any of the keyword names to name variables

Variables

- Variable names correspond to locations in the computer's memory
  - Every variable has a name, a type, a size and a value
  - Reading variables from memory does not change them

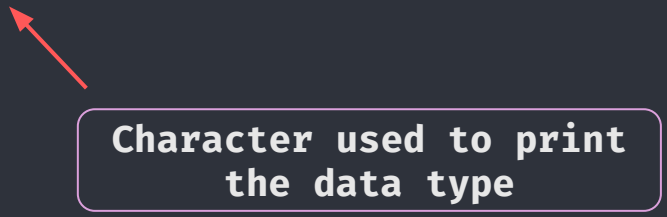


Variables


Variable Declaration
int number ;
number = 10;
Variable Type
Variable Name
Variable Value
Variable Initialization

Display variables

```c
#include <stdio.h>
int main() {
    int a; a = 10;
    printf("The value of a is = %d\n", a);
    return 0;
}
```



https://www.geeksforgeeks.org/printf-in-c/

Data Types in C

```c
include <stdio.h>
int main() {
     int num = 10; // Integer data type
     printf("Integer: %d\n", num);
     float floatNum = 3.14; // Floating-point data type
     printf("Float: %.2f\n", floatNum);
     char character = 'A'; // Character data type
     printf("Character: %c\n", character);
     _Bool boolVar = 1; // Boolean data type
     printf("Boolean: %d\n", boolVar);
     unsigned int unsignedNum = 100; // Unsigned integer data type
     printf("Unsigned Integer: %u\n", unsignedNum);
     long longNum = 1234567890; // Long integer data type
     printf("Long Integer: %ld\n", longNum);
     double doubleNum = 2.71828; // Double data type
     printf("Double: %.5f\n", doubleNum);
     return 0;
     }
```

Data Types in C

In C programming, data types are declarations for variables. This determines the type and size of data associated with variables. For example,

```
int myVar;
```

Here, myVar is a variable of int (integer) type. The size of int is 4 bytes.

Basic Data Types

Data Type	Size (bytes)	Format Specifier
int	4	%d, %i
char	1	%c
float	4	%f
double	8	%lf

Data Types in C

int - Integers are whole numbers that can have both zero, positive and negative values but no decimal values.

For example, 0, -5, 10

float and double are used to hold real numbers.

For example, 10.5, 22.9.

Char - Keyword char is used for declaring character type variables.

For example: char test = 'h';

```
Thanks; {

    ‘Do you have any questions?’

        < bgamage@sjp.ac.lk >

}
```