

# C Programming Basics

SSC 322/329

Email any questions to:  
[rauta@tacc.utexas.edu](mailto:rauta@tacc.utexas.edu)

# Overview of Content

- Writing a Basic C Program
- Understanding Errors
- Comments, Keywords, Identifiers, Variables
- Operators
- Standard Input and Output
- Control Structures
- Functions in C
- Arrays, Structures
- Pointers
- Working with Files

**All the concepts are accompanied by examples.**

# Creating a C Program

- Have an idea about what to program
- Write the source code using an editor or an Integrated Development Environment (IDE)
- Compile the source code and link the program using a C compiler
- Fix errors, if any
- Run the program and test it
- Fix bugs, if any

# Write the Source Code: firstCode.c

```
#include <stdio.h>
```

```
int main() {
```

```
    printf("Introduction to C!\n");
```

```
    return 0;
```

```
}
```

Output:

Introduction to C!

# Understanding firstCode.c

Preprocessor directive

#include <stdio.h> → Name of the standard header file to be included is specified within angular brackets

Function's return type

Function name

Function name is followed by parentheses – when empty no arguments are being passed

int main ( ) {

printf ( "Introduction to C! \n" ) ;

C language function for displaying information on the screen

return 0 ;

Keyword, command for returning function value

}

→ The contents of the functions are placed inside the curly braces

Text strings are specified within " " and every statement is terminated by ;

Newline character is specified by \n

# Save-Compile-Link-Run

- Save your program (source code) in a file having a “c” extension.

Example, `firstCode.c`

- Compile and Link your code (by default, GCC automatically does the linking)

```
gcc -o firstCode firstCode.c
```

- Run the program

```
./firstCode
```

Repeat the steps above every time you fix an error!

# Different Compilers

- Different commands for different compilers (e.g., **icc** for intel compiler and **pgcc** for pgi compiler)

- GNU C program

```
gcc -o firstCode firstCode.c
```

- Intel C program

```
icc -o firstCode firstCode.c
```

- PGI C program

```
pgcc -o firstCode firstCode.c
```

- To see a list of compiler options, their syntax, and a terse explanation, execute the compiler command with the -help or --help option

# Summary of C Language Components

- Keywords and rules to use the keywords
- Standard header files containing functions like **printf**
- Preprocessor directives for including the (standard) header files
- Function **main**
- Parentheses and braces for grouping together statements and parts of programs
- Punctuation like **;**
- Operators like **+**
- All the above and more to come make up the syntax of C



# Pop Quiz

(add the missing components)

```
_____ <stdio.h>

int main()____

    printf("Introduction to C!\n") ____

    printf("This is a great class!\n");

    return 0;
```

# References

- C Programming Language, Brian Kernighan and Dennis Ritchie
- Let Us C, Yashavant Kanetkar
- C for Dummies, Dan Gookin
- <http://cplusplus.com>
- <http://www.cprogramming.com/tutorial/c/lesson11.html>