

Structure of a C Program

ESC108A Elements of Computer Science and Engineering
B. Tech. 2017

Course Leaders:

Roopa G.

Ami Rai E.

Chaitra S.



Objectives

- At the end of this lecture, student will be able to
 - Explain the structure of a C program
 - Explain the building blocks of C programs

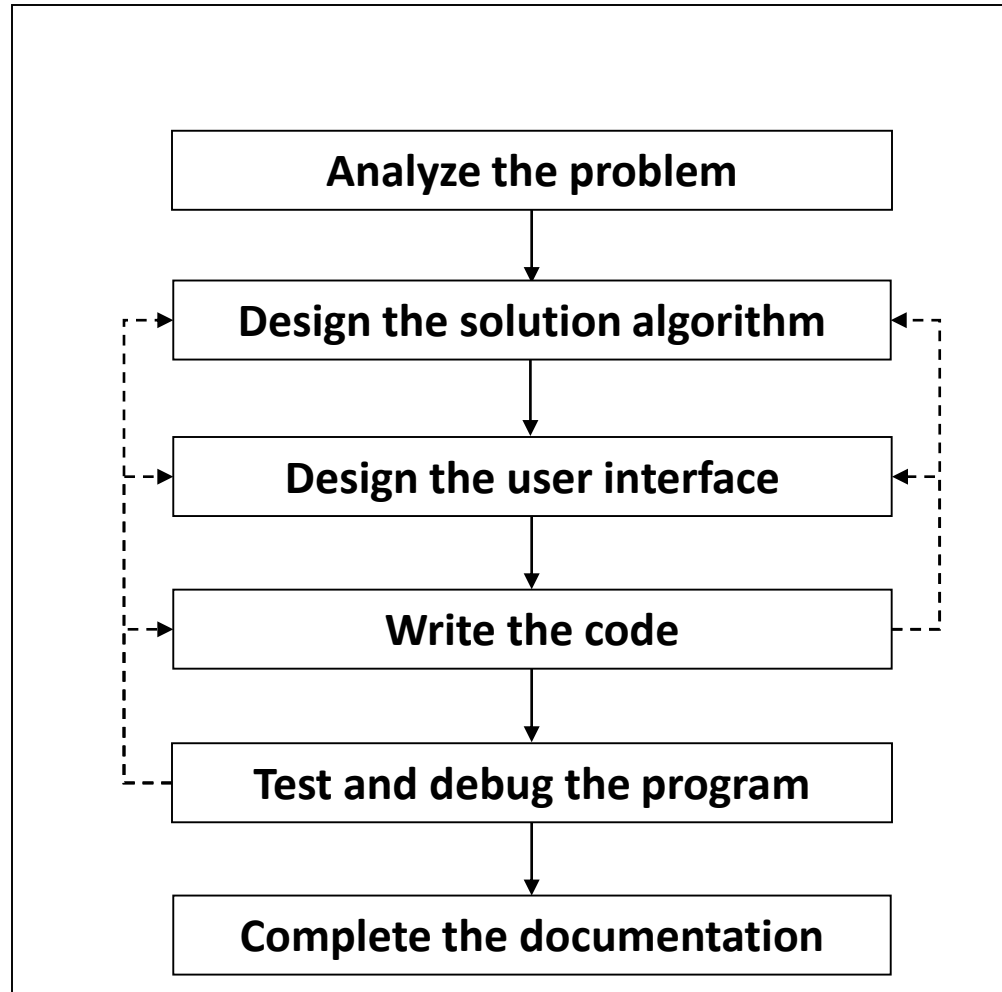


Contents

- Introduction to C programming
- Tokens, keywords, identifiers and constants
- Expressions, Blocks and Statements



Program Development Cycle

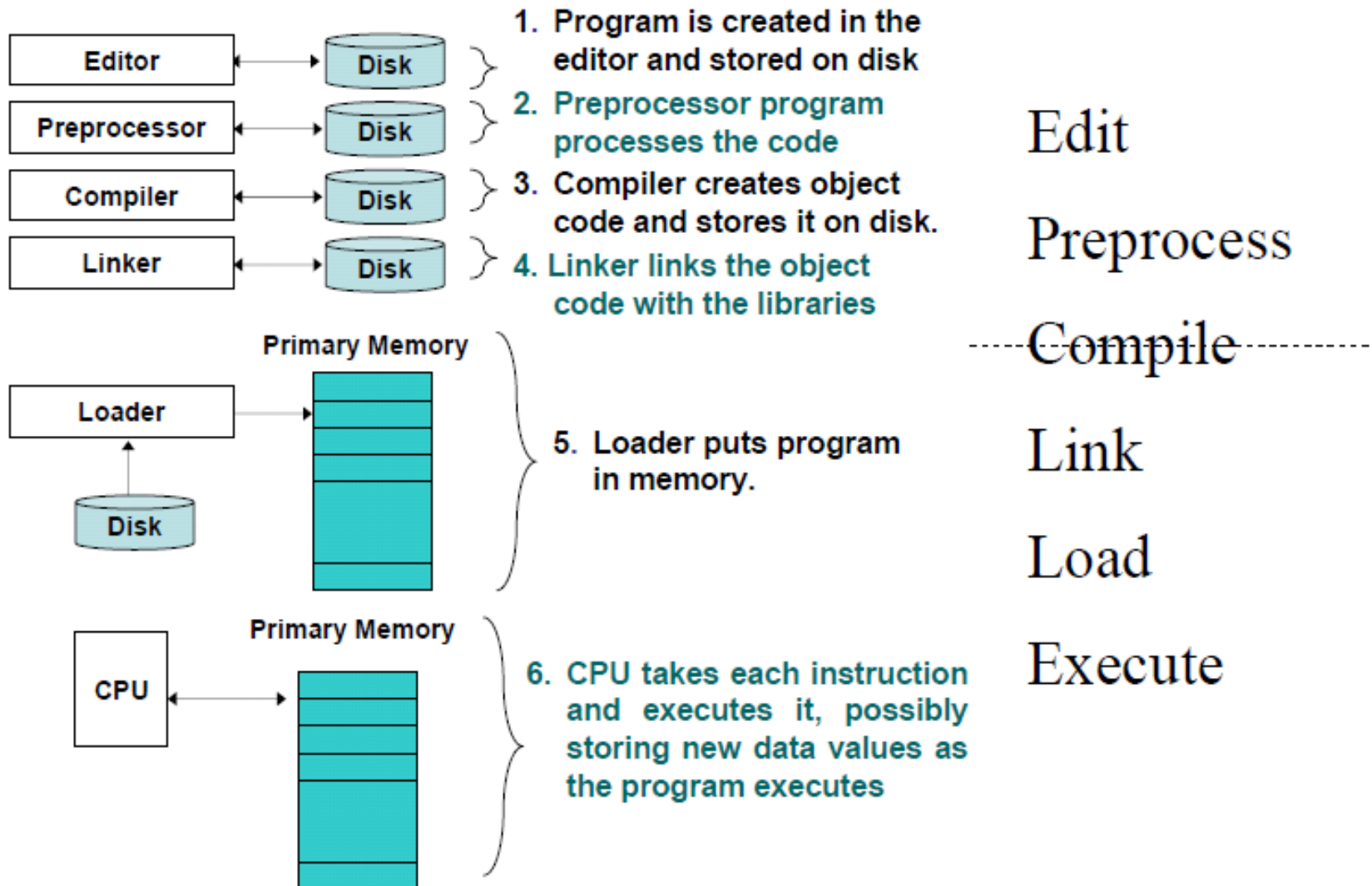


C Language

- Developed by **Dennis Ritchie** at Bell Laboratory
- Procedure oriented language
- Evolved from B, which evolved from BCPL



Phases of C Programs



Structure of a C Program

Documentation section

link section

definition section

Global declaration section

main() function

Braces

statements

.....

user defined functions

```
/*This is a program to calculate perimeter  
of circle - vsarma*/
```

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

```
#define PI 3.14
```

```
float area(float);
```

```
int main(int argc, char *argv[]){
```

```
    printf("Perimeter of circle with radius  
    %f is %f\n",2.0f,area(2.0));
```

```
}
```

```
float area(float radius){
```

```
    return 2*PI*radius;
```

```
}
```



main() Function

- **main()** is a part of every C program
- C programs contain one or more functions, one of which must be main
- Every program in C begins executing at the main()
- **Block** – the pair of braces {} and the portion of program between the braces
- **Statement terminator** – every statement must end with a ;



Question

- We all know one or more human languages. What is a human language made up of?
 - Alphabets
 - Words
 - Sentences
 - Paragraphs
- A computer program
 - Character set
 - Tokens
 - Statements
 - Functions



Character Set

- Alphabets

- Lower case letters – a to z
- Upper case letters – A to Z

- Digits

- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

- Special characters

- ~ % | @ + < _ - > ^ # = & \$ / (* \) ' : [" ;] ! , { ?
. }



C Tokens

- C tokens : the smallest individual units
 - Keyword – float, while, for, int,....
 - Identifier – main() , amount, sum, ...
 - Constants – -13.5, 500, ...
 - Strings – “ABC”, “MCA”, ...
 - Operators – + - * % ...
 - Special Symbols – [] { }...



Keywords used in C

- Fixed meaning, cannot be changed
- Reserved words
- Cannot be used as a variable or function name
- Basic building blocks
- All in Lowercase



Keywords used in C contd.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while



Identifiers

- Name given to a function or variable memory location
- An identifier is a series of characters consisting of letters, digits and underscores “_”
- Should start with a letter or underscore
- Can be any length
 - only the first 31 characters are required to be recognized by ANSI C compilers
 - Keep identifiers 31 characters or less for portability and fewer problems



Identifiers - Examples

- CAN
 - contain a number elsewhere `h2o`
 - be of mixed cases `Xsquared`
 - contain or begin with an underscore `_height_`
- CANNOT
 - start with a number `2i`
 - contain any arithmetic operators `r*s+t`
 - contain any other punctuation marks `#@x%£!!a`
 - be a C keyword `struct`
 - contain a space `my var`



Constants

- Fixed values
- Does not changes during execution of program
- Numeric constant – Integer (decimal, octal, hexadecimal) and Real
- Character constant :
 - Single character constant
 - String constant
 - Backslash character constant



Escape Sequence

- Escape character – backslash \
- When encountering a backslash in a string, the compiler looks ahead at the next character and combines it with \ to form **escape sequence**



Common Escape Sequences

Escape sequence	Description
\' (single quote)	Output the single quote (') character.
\" (double quote)	Output the double quote (") character.
\? (question mark)	Output the question mark (?) character.
\\ (backslash)	Output the backslash (\) character.
\a (alert or bell)	Cause an audible (bell) or visual alert.
\b (backspace)	Move the cursor back one position on the current line.
\f (new page or form feed)	Move the cursor to the start of the next logical page.
\n (newline)	Move the cursor to the beginning of the next line.
\r (carriage return)	Move the cursor to the beginning of the current line.
\t (horizontal tab)	Move the cursor to the next horizontal tab position.
\v (vertical tab)	Move the cursor to the next vertical tab position.



Summary

- All C programs follow a predefined structure
- Words of a computer programming language are known as Tokens
 - They can be Reserved Keywords, Identifiers, Operators and Constants
- Expressions are a collection of operands and operators
- Statements are always terminated
- Statements can be composed of declarations, expressions, control structures or a function call



Further Reading

Allain, A. (2005) *Introduction to C*, available at <http://www.cprogramming.com/tutorial/c/lesson1.html> (accessed 22 July 2014).

Kernighan, B. W. and Richie, D. (1992) *The C Programming Language*. 2nd ed., New Delhi:PHI.

