



Arab Academy for Science and Technology and Maritime Transport

College of Engineering and Technology

Computer Engineering Department

CC112 Structured Programming

Lecture 2

LECTURE 2

Data Types Assignments Statements

LECTURE OUTLINE

- i. **C Program Structure**
- ii. **C Language Elements**
 - Preprocessor directives
 - Main Function Header
 - Comments
 - Statements
 - Variable declaration
- iii. **Data Types and Variable Declarations**
- iv. **Assignments in C**

i. C PROGRAM STRUCTURE

Preprocessor directives

Main function prototype

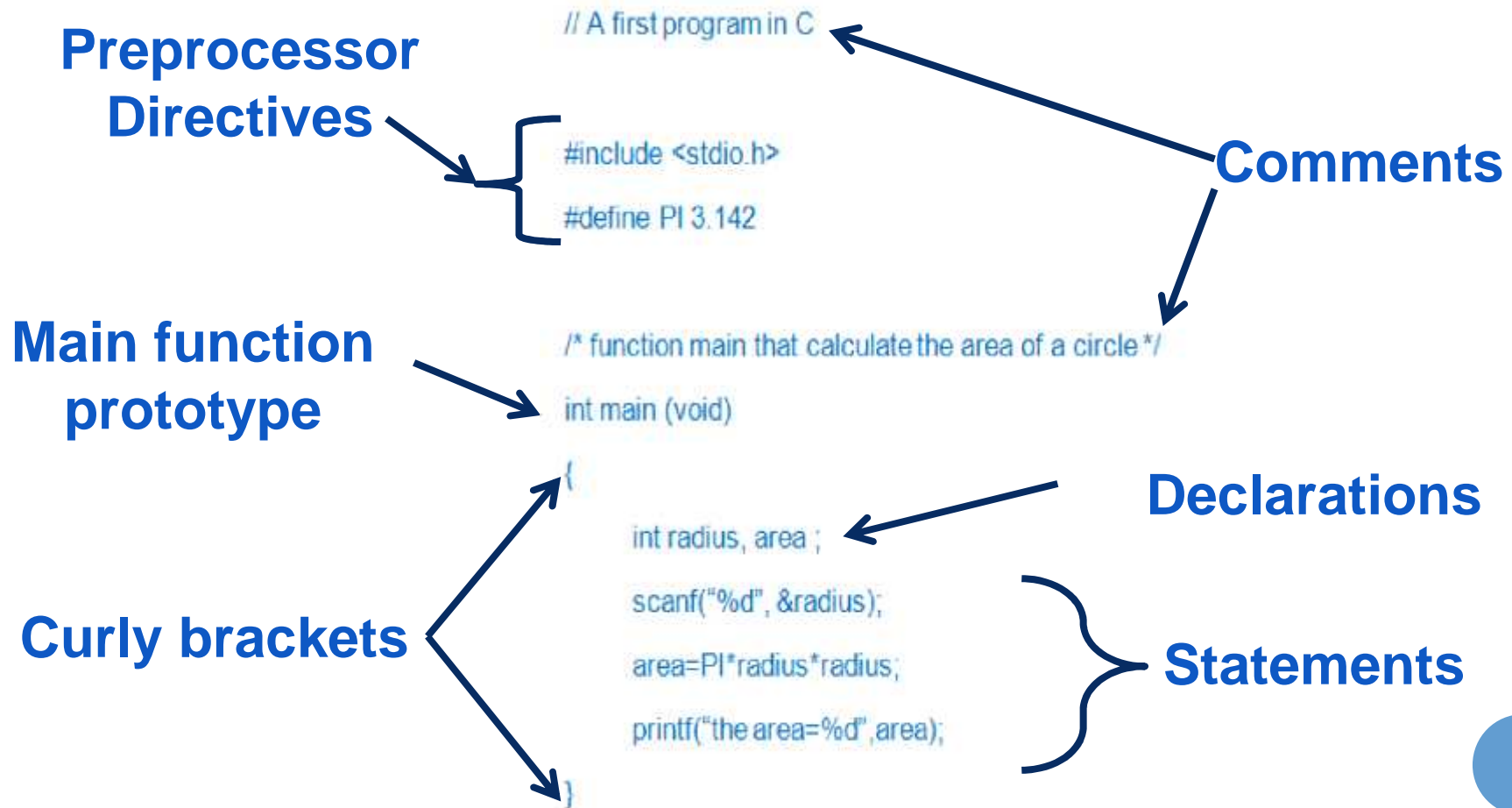
{

Declarations

Statements

}

i. C PROGRAM STRUCTURE



II. C LANGUAGE ELEMENTS

Preprocessor Directives

- Are commands that give instructions to the C preprocessor
- Begins with a (#) as its nonblank character.
- C preprocessor is a system program that modifies a C program prior to its compilation.

Example1

#include <stdio.h>

- The C language cannot do I/O by itself, so we need help from the library “stdio.h” to use the screen/Keyboard !
- We can use other libraries, too , as needed.
- Another popular “library” is math.h , for advanced math functions.

II. C LANGUAGE ELEMENTS

Preprocessor Directives

Example 2

define PI 3.142

- Define is a preprocessor directives.
- Valid constant declarations
- A named constant is a location in memory that we can refer to by a name, and in which a data value that cannot be changed is stored.
- This directives instructs the processor to replace each occurrence of PI by 3.142

II. C LANGUAGE ELEMENTS

Main Function Prototype

The main function of a simple C program is:

```
int main(void)  
{  
}
```

This is where the program execution begins.

The curly braces mark the beginning and end of the function.

II. C LANGUAGE ELEMENTS

Comments

- You insert comments to programs in order to improve program readability.
- Comments also help other people read and understand your program.
- Comments do not cause the computer to perform any action when the program is run.
- Begin with `/*` and ends with `*/`
- Or begins only with `//`

II. C LANGUAGE ELEMENTS

Statements

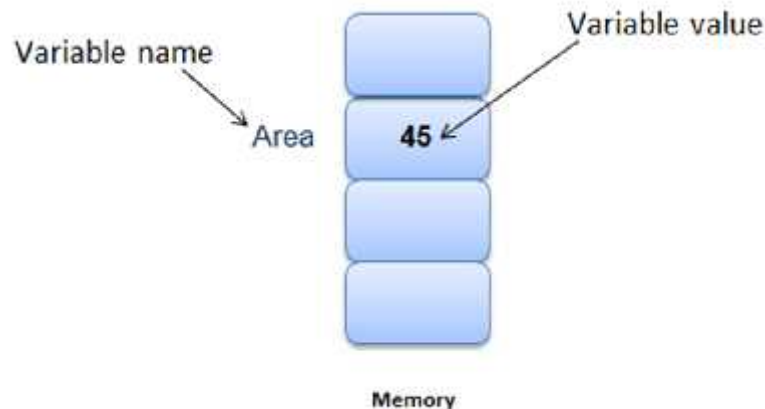
- These statements appear in the program body just after the declarations.
- Their purpose is to perform operations such as reading input from the user, displaying output, perform arithmetic operations, etc...
- Every statement must end with a semicolon (also known as the statement terminator).

II. C LANGUAGE ELEMENTS

Declarations

- **Variable**

- A name associated with a memory cell whose value can change.
- Variables are used to store a program's input data and its computational results.



II. C LANGUAGE ELEMENTS

Declarations

- **Variable Declaration**

Statements that communicate to the compiler the names of variables in the program and the type of information stored in each variable.

```
int area;
```

II. C LANGUAGE ELEMENTS

Declarations

- o **Variable Declaration Rules**

A variable name in C is any valid **identifier**.

An identifier is a series of characters consisting of letters, digits and underscores (`_`) that does *not* begin with a digit.

C is **case** sensitive—uppercase and lowercase letters are different in C, so `a1` and `A1` are different identifiers.

C reserved words and standard identifiers cannot be used.

II. C LANGUAGE ELEMENTS

Declarations

o Reserved Word

A word that has special meaning in C such as:

- **int**
- **void**
- **double**
- **return**

o Standard Identifiers

A word having special meaning but one that a programmer may redefine (but redefinition is not recommended)

- **printf**
- **scanf**

II. C LANGUAGE ELEMENTS

Declarations

o Invalid identifiers

1Letter	begins with a number
double	reserved word
int	reserved word
TWO*FOUR	character * not allowed
joe's	character ' not allowed
age#	character # not allowed
Age-of-cat	character – is not underscore character (_)

o Valid Identifiers

Age_of_person
taxRateY2k
PrintHeading

III. DATA TYPES AND VARIABLE DECLARATIONS

o Integral Types

- Whole numbers declared as **int**, **short**, or **long**
- **Int sample values** 4578, -4578, 0

o Floating Types

- Real numbers, declared as **float**, or **double**
- **Float sample values** 95.274 95.0 0.265

o Character Types

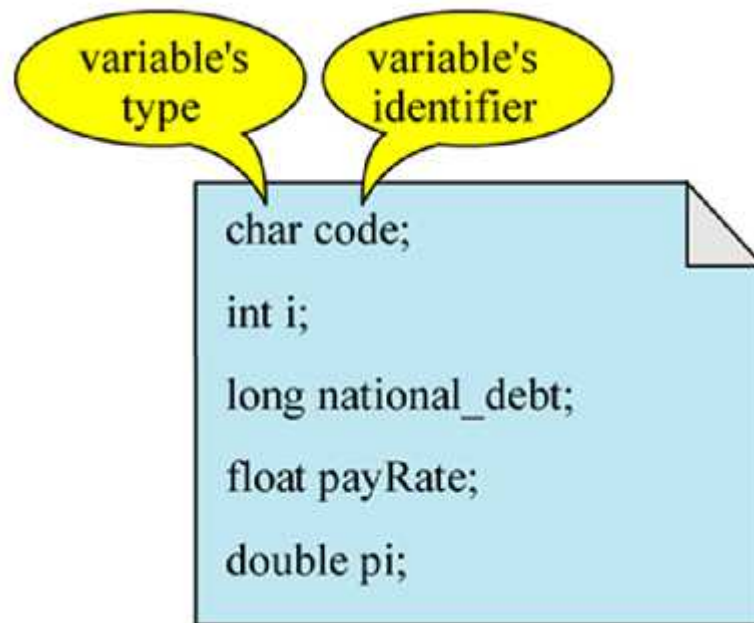
- Single characters, declared as **char**
- **Char sample values** B d 4 ? *

III. DATA TYPES AND VARIABLE DECLARATIONS

TYPE	DESCRIPTION	EXAMPLE	DECLARATION
Integral	Whole numbers	4578, -4578, 0	int, or short
Floating	Real numbers	95.27, -95.0	float, or double
Character	Single characters	B d 4 ? *	char

III. DATA TYPES AND VARIABLE DECLARATIONS

VARIABLE DECLARATIONS EXAMPLE



IV. ASSIGNMENTS IN C

- The assignment is to store in a variable the results of a computation.
- The symbol **=** assignment operator in C.
- Example:

```
code = 'B';  
i = 14;  
a=b*c+d;
```

THANK YOU