Faculty of Computers & Informatics

Introduction to Programming

Lecture 3: Basic Data Types & Operators

Mahmoud El-Gayyar

elgayyar@ci.suez.edu.eg

Review Chapter 2

- *Introduction to C Langauge*
  - Tool
  - Dangerous
- *Examples of C Programs*
- *Program Structure*

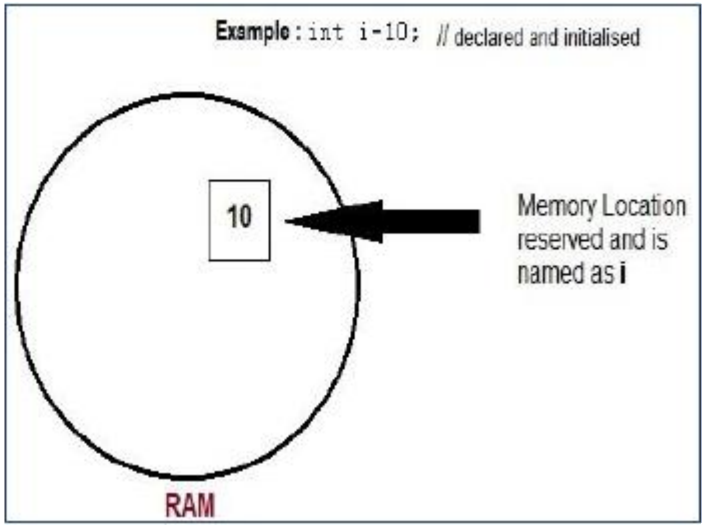
Outline

- *Types*
- *Constants*
- *Declarations*
  - Variable Names
- *Operators*
  - Arithmetic Operators
  - Assignment Operators
- *Function Calls*

Outline

- ***Types***
- *Constants*
- *Declarations*
  - Variable Names
- *Operators*
  - Arithmetic Operators
  - Assignment Operators
- *Function Calls*

Types


Example : int i-10; // declared and initialised
10
Memory Location reserved and is named as i
RAM

Type	No. Of bits	Minimum Value Range
char	8	0-256
int	16/32	-32,767 to 32,767
longint	32/64	-2,147,483,647 to +2,147,483,647
float	32	A floating-point number
double	64	A floating-point number with more precision

Outline

- *Types*
- ***Constants***
- *Declarations*
  - Variable Names
- *Operators*
  - Arithmetic Operators
  - Assignment Operators
- *Function Calls*

Constants

- *Integers*
  - 10, 0, -10, 10L
- *Floating Points (decimal point or symbol **e**)*
  - 3.14, 10., .01, 123e4, 123.456e7
  - (e) is the power of 10
  - Double by default
- *Characters and Strings*
  - ‘a’, ‘$’, ‘1’, “hello”
  - “\”hi\”\t\\\’man\’”

User-defined Constants

```
#define PI 3.14156

#define MYNAME "JOHN DOE"

#define LIMIT 10
```

- *Statements started by # are called* ***preprocessor directive***
- *Top of the program (after include), single line for each*
- *Why this can be very important?!!*

Outline

- *Types*
- *Constants*
- ***Declarations***
  - Variable Names
- *Operators*
  - Arithmetic Operators
  - Assignment Operators
- *Function Calls*

Delcarations

- *A declaration tells the compiler the **name** and **type***

```
char c;
```

- *A declaration for a variable can also contain an **initial value.***

```
int i=10;
```

- *You can also declare several variables of the same type in one declaration, separating them with commas*

```
float f1=10, f2=20;
```

Variable Names (identifiers)

- *The first character must be a letter, either lowercase or uppercase*
- *Afterwards, you can use letters, digits or underscore*
- *C is case-sensitive; reserved words are not allowed (e.g. int)*
- *The variable must be unique in the* ***first eight characters*** *in order to be safe across compilers*
- ***2nd Good Programming Style:***

  - Defined constants are traditionally made up of all uppercase characters
  - Make variable names descriptive (e.g. name it salary not s)
  - Separate between different words:
    - *theSum, the_sum*

Outline

- *Types*
- *Constants*
- *Declarations*
  - Variable Names
- ***Operators***
  - Arithmetic Operators
  - Assignment Operators
- *Function Calls*

Arithmetic Operators

Arithmetic Operator	Operands	Function
-	Unary	Sign(negatea number)
%	Binary	Modulus(remainder)
/	Binary	Division
*	Binary	Multiplication
-	Binary	Subtraction
+	Binary	Addition

Examples: Arithmetic Operators

1/2	→	*0*	
5/2	→	*2*	
1 / 2.0	→	*0.5*	
1 % 10	→	*1*	
11 % 3	→	*2*	
*1+2*3*	→	*1 + (2*3)*	→ *7*
*(1 + 2) * 3*	→	*9*	
1.5 + 2 / 3 + 2.0	→	*1.5+(2 / 3)+2.0*	→ *3.5*

Assignment Operators

x=1	→	*sets x to 1*
a=b	→	*assigns the current b value to a*
i=i+1	→	*increment i by 1*
c=a=b	→	*c=(a=b)*

Outline

- *Types*
- *Constants*
- *Declarations*
  - Variable Names
- *Operators*
  - Arithmetic Operators
  - Assignment Operators
- ***Function Calls***

Function Calls

- ***Built-in Functions:*** *piece of code written by someone else to perform some useful task.*
- *To use a built in function, you need to call it*
  - function_name (arguments) // arguments are separated by commas (,)
  - The function may return a result // you need to store it in ?!!
- *Examples:*
  - printf("Hello, world!\n")
  - printf("%d\n", i)
  - sqrt(144) ⚠
  - printf("sum = %d\n", a + b + c) // arguments can be expressions
  - c = sqrt(a * a + b * b)

Summary

- *How to declare variables of different types?*
  - Always use descriptive names for your variables
- *Use the #define directive to define constants*
- *Division of two integers is always integer*
- *Use explicit () to change operators precedence*
- *How to call functions?*
  - Don’t forget to store its return result.