DATA TYPES AND VARIABLES

UNIVERSITY of KIGALI

The c character set

1. **Letters(Both uppercase letters & lowercase letters):** Capital **A** to **Z**

   Small **a** to **z**

2. **Digits :** All decimal digits **0** to **9**

3. **White spaces:** Blank space(\)

   New line(\n)

4. **Special characters:** # % & ! _ {} [] () < > | + - / * =,etc...

UNIVERSITY Of KIGALI

Special characters

’	Comma	&	Ampersand
.	Period or dot	^	Caret
;	Semi-colon	*	Asterisk
:	Colon	-	Minus
`	Apostrophe	+	Plus
``	Quotation mark	<	Less than
!	Exclamation mark	>	Greater than
|	Vertical bar	( )	Parenthesis left / right
/	Slash	[ ]	Bracket left / right
\	Back slash	{ }	Braces left / right
~	Tilde	%	Percent
–	Underscore	#	Number sign or Hash
$	Dollar	=	Equal to
?	Question Mark	@	At the rate

UNIVERSITY of KIGALI

The C-Programming keywords:

- The **C** keywords are reserved words by the compiler.
- All the **C** keywords have been assigned fixed meaning.
- The keywords can not be used as variables names because they have been assigned fixed jobs.

UNIVERSITY Of KIGALI

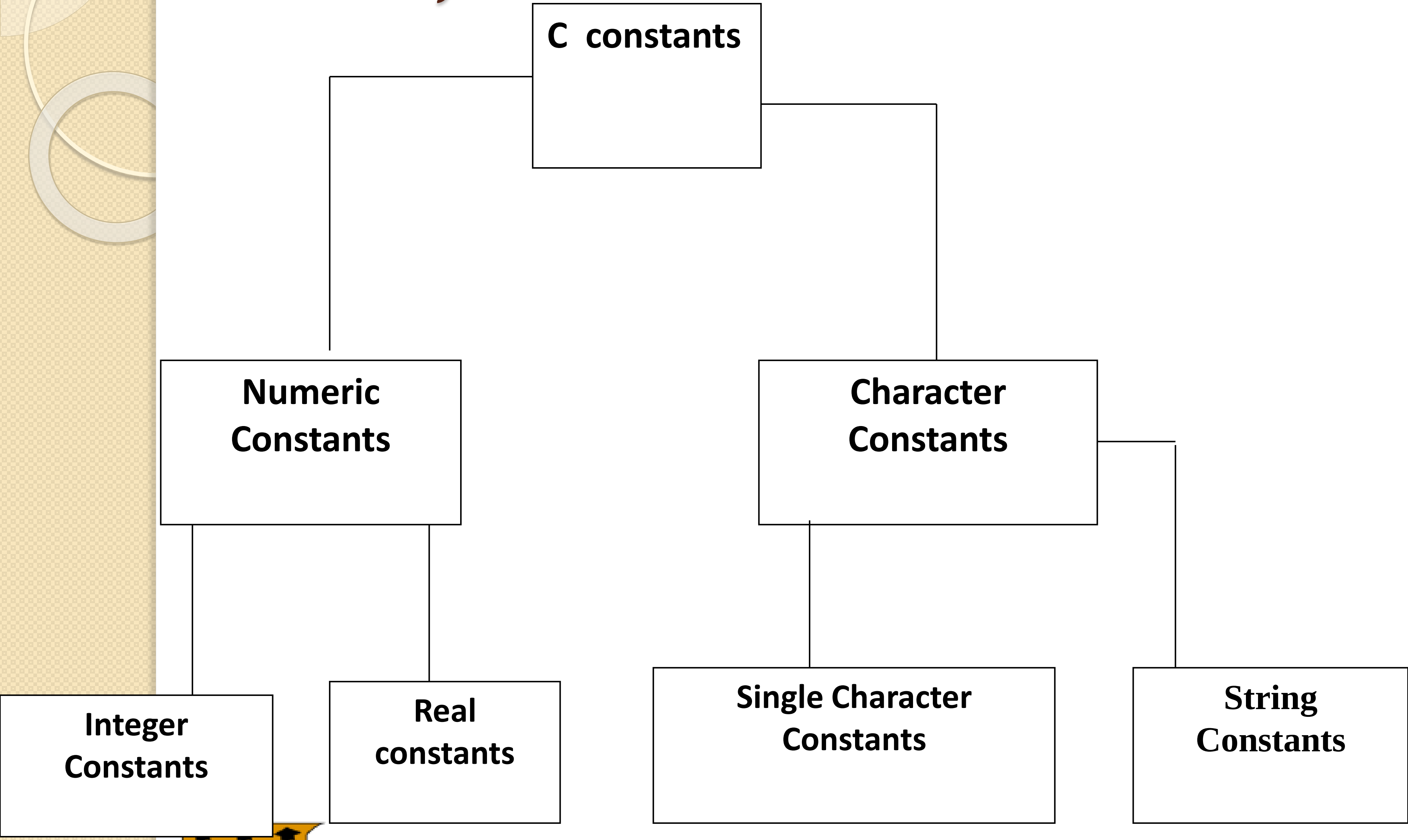
keywords

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

Constants:

- The constants in **C** are applicable to the **values**, which do not change during the execution of a program.
- There are several types of constants in C.

Constants, …



Real Constants Real Constants String Constants String Constants

A. Numerical Constants

- ***1) Integer constants***
- These are the sequence of numbers from 0 to 9 without decimal points or fractional part or any other symbols.
- Integer constants could either be positive or negative or may be zero. The number without a sign is assumed as positive.
- **Example** 10, 20, +30, -44 etc

2) Real constants

- Real constants are often known as ***floating point constants.*** Many parameters or quantities are defined not only in integers but also in real numbers.
- For example, length, height, prize etc. are measured in real numbers.
- **Example** 2.5, 5.521 etc

B. Character Constant

1) Single character constants

- A character constant is a single character. They are also represented with a single digit or a single special symbol or white space enclosed within a pair of single quote marks.
- **Example** 'a', '8', '' etc.

2) String constants

- String constants are sequence of characters enclosed within a double quote marks. The string may be a combination of all kinds of symbols.
- **Example** "Hello", "a", "UK"

UNIVERSITY
Of KIGALI

Symbolic constants

- A symbolic constant is a constant that is represented by a name(symbol) in your program.
- Like literal constant, a symbolic constant can't change.
- Whenever you need the constant's value in your program, you use its name as you would use a variable name.
- The actual value of the symbolic constant needs to be entered only once, when it first defined.

UNIVERSITY Of KIGALI

Defining a symbolic constant

- C has two methods for defining a symbolic constant:
- The #define directive is used as follows:

- **The first one is in this format**

#define CONSTANTNAME literal

- ex: #define PI 3.14159

- By convention the names of the symbolic constant are uppercase, this make them easy to distinguish from variables names which by convention are lowercase.

UNIVERSITY Of KIGALI

VARIABLES

- **A variable** is just a named storage area in the computer's memory that can hold a single value (numeric or character).
- **A variable** name may be declared based on the meaning of the operation. Some meaningful variable names are as follows.
- **Example** height, average, sum etc.

Rules for defining variables

- They must begin with a character without spaces but underscore is permitted.
- The length of the variable varies from compiler to compiler. Generally most of the compilers support 8 characters. However, the other standard recognizes the maximum length of a variable up to 31 characters.
- The variable should not be a C keyword
- The variable names may be a combination of upper and lower characters. For example suM and sum are not the same.
- The variable name should not start with a digit

UNIVERSITY Of KIGALI

DATA TYPES

- The data types are integers, real or character constants.
- **1. Integers Data Type**
- ***a) <u>Integer, short and long</u>***

- All C compilers offer different integer data types.
- They are short and long.
- Short integer requires half the space in the memory than the long one.
- The **short integer** requires <u>two bytes</u> and the **long integers** <u>four bytes</u>.

Difference between short and long integers

Short integer	Long integer
Occupies 2 bytes in memory	Occupies 4 bytes in memory
Range: -32 768 to 32 767 using Turbo++ and 320M using Dev++	Range: - 2 147 483 648 to 2 147 483 647
Program runs faster	Program runs slower
Control string is **%d** or **%I**	Control string **%ld**
Example: **int a=2;** **short int b=2;** When a variable is declared without short or long keyword, the default is short-signed int.	Example: **long b;** **long int c;**

UNIVERSITY of KIGALI

b) *Integers, signed and unsigned*

Difference between signed and unsigned integers

Signed integer	Unsigned integer
Occupies 2 bytes in memory	Occupies 2 bytes in memory
Range: -32 768 to 32 767	Range: 0 to 65 535
Control string is **%d** or **%I**	Control string **%u**
By default signed int is short-signed int.	By default unsigned int is short unsigned int.
There are also long signed integer having range from - 2 147 483 648 to 2 147 483 647	There are also long unsigned int with range 0 to 4 294 967 295

Example:
int a=2;
short int b=2;
	Example:
	unsigned long b;
	unsigned long int c;

2. Char, Signed and Unsigned

Difference between signed and unsigned char

Signed char	Unsigned char
Occupies 1 byte in memory	Occupies 1 byte in memory
Range: -128 to 127	Range: 0 to 255
Control string is **%c**	Control string :**%c**

UNIVERSITY OF KIGALI

3. Floats and Doubles

Difference between float and double

Float	Double
Occupies 4 bytes in memory	Occupies 8 bytes in memory
Range: $-3.4e10^{-38}$ to $+3.4e10^{+38}$	Range: $-1.7e10^{-308}$ to $+1.7e10^{+308}$
Control string is **%f**	Control string :**%lf**
Example: **Float a;**	Example: **Double y;** There also exist **long double** having ranged $-1.7e10^{-4\,932}$ to $+1.7e10^{+4\,932}$ and occupies 10 bytes in memory. Example: **long double k;**

UNIVERSITY OF KIGALI

The entire data types supported by the 'C'.

Data type	**Size (bytes)**	**Range**	**Control String**
Char	1	-128 to 127	**%c**
Unsigned char	1	0 to 255	**%c**
Short or int	2	- 32 768 to 32 767	**%d or %i**
Unsigned int	2	0 to 655 355	**%u**
Long	4	-2 147 438 648 to2 147 438 647	**%ld**
Unsigned long	4	0 to 4 294 967 295	**%lu**
Float	4	-3.4e10^{-38} to +3.4e10^{+38}	**%f or %g**
Double	8	-1.7e10^{-308} to +1.7e10^{+308}	**%lf**
Long double	10	-1.7e10$^{-4\,932}$ to +1.7e10$^{+4\,932}$	**%lf**

Declaring variables

- done in the declaration part of the program.
- The variables must be declared before they are used in the program.
- **Declaration provides two things:**
- Compiler obtains the **variable name**.
- It tells to the compiler **data type of the variable** being declared and helps in allocating the memory.

- The syntax of declaring a variable is as follows:

- Syntax
- **Data type   variable_name;**

- **Example:**
- *int age;   /* integer */*
- *char m; /* character */*
- *float s;   /* float */*
- *double k; /* double */*
- *int a,b,c;*

Initializing variables

- Variables declared can be assigned or initialized using an assignment operator '**=**'.
- The declaration and initialization can also be done in the same line.
- In its simplest form, a declaration consists of the type, the name of the variable, and a terminating semicolon

Syntax:

data_type variable_name = constant

Example:

- **Int x=1; where x is an integer variable.**
- **int y=2;**
- A **declaration** tells the compiler the name and type of a variable you'll be using in your program

UNIVERSITY Of KIGALI

Declarations and Initialization

- ***Initialization** simply mean assign a value to a variable*
- **Int i = 0;**
- **char ch = 'a';**
- are equivalent to the more longwinded
- **int i;**
- **char ch;**
- **i = 0;**
- **ch = 'a';**

Declarations of different variables of the same data type

- To declare a variable in a C program one writes the *type* followed by a list of variable *names* which are to be treated as being that type:
- **data Type name *variablename1,..,..,variablenameN***;
- For example:
- int i,j;
- char ch;
- double x,y,z,fred;

UNIVERSITY Of KIGALI

Where to declare things

There are two kinds of place in which declarations can be made

1. One place is outside all of the functions.

- That is, in the space between function definitions. (After the #include lines, for example.)
- Variables declared here are called **global variables**.
- There are also called static and external variables in special cases.)

2. The other place where declarations can be made is following the opening brace, {}, of a block.

- Variables of this kind only work inside their braces {} and are often called **local variables.**

UNIVERSITY
Of KIGALI