VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CSE1002
Introduction to C Programming

Dr.B.Saleena

School of Computer Science and Engineering

VIT, Chennai Campus

About the course

- A lab only course for problem solving and coding skill development

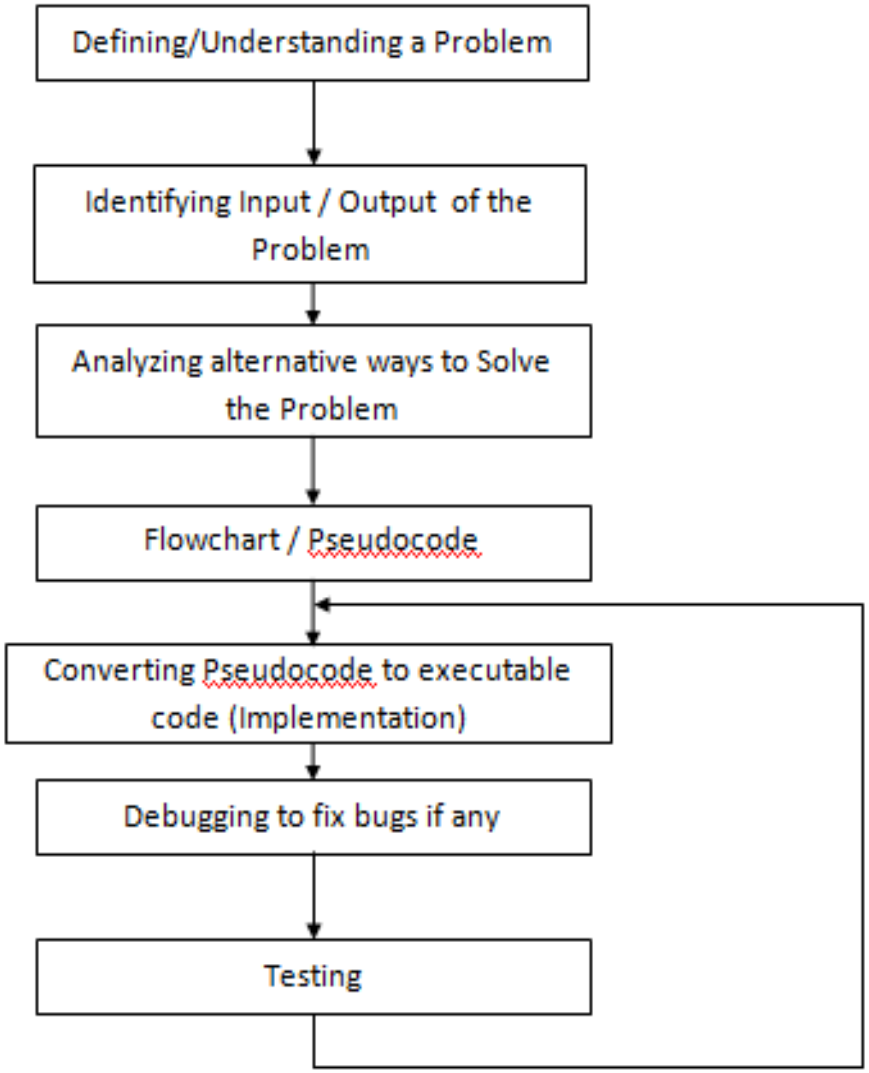
Few Links

- https://www.tutorialspoint.com/cprogramming/
- https://www.w3schools.com/cpp/default.asp
- https://www.w3schools.in/c-tutorial/

Evaluation

- **Periodic Assessment Test (4)---8%**
- **Consolidated Assessment Test (2) ---14%**
- **Inlab Practice Session**

Problem Solving steps- A Recap



Python

- Easy to learn
- Language with Simple rules
- Good for beginners
- Code is readable

Difference between Python and C

- **C programs – Compiled**
- **Python programs – Interpretered**

Compiler	Interpreter
Takes **entire** program as input and generate a output file with object code	Takes instruction by instruction as input and gives an output. But does not generate a file
Errors are displayed after **entire program** is checked	**Errors** are displayed for **every instruction** interpreted (if any)

History of C

- Born at AT & T Bell Laboratory of USA in 1972
- Many of C's principles and ideas were derived from the earlier language B
- Ken Thompson was the developer of B Language
- C was written by Dennis Ritchie

  http://www.nytimes.com/2011/10/14/technology/dennis-ritchie-programming-trailblazer-dies-at-70.html?_r=0
- C language was created for a specific purpose i.e designing the UNIX operating system (which is currently base of many UNIX based OS)
- Quickly spread beyond Bell Labs in the late 70's because of its strong features

Features of “C”

Middle level language:

- “C” has features of both assembly level language (Low level language) & High level language

Structured Programming:

- is a technique in programming language
- divide a big program into number of small programs

Features of “C” – contd..

“C” is extensible:

- “C” compiler comes with standard library
- Standard library is a low level codes for built-in functions
- the same way, user defined functions can also be added to the library and make it available for others to use.

Features of “C” – contd..

“C” is Portable:

- language which written on one computer can run on others without or with less modifications.

C is rich in data types and operators

Structure of a C program

Documentation Session – Comments
pre-processor directives – Preceded by a '#'
global declarations
int main() - standard start for all C programs
{
local variables to function main ; - all variables used in the functio
must be declared in the beginning
statements associated with function main ;
}
void f1()
{
local variables to function 1 ;
statements associated with function 1 ;
}

Components of a C program

A C program consists of the following parts:

- Comments
- Variables
- Preprocessor Commands
- Functions
- Statements & Expressions

Comments in C

Two types of comments

Single line and multi line comment

Single Line Comment is double forward slash ‘//’ and can be **Placed Anywhere**

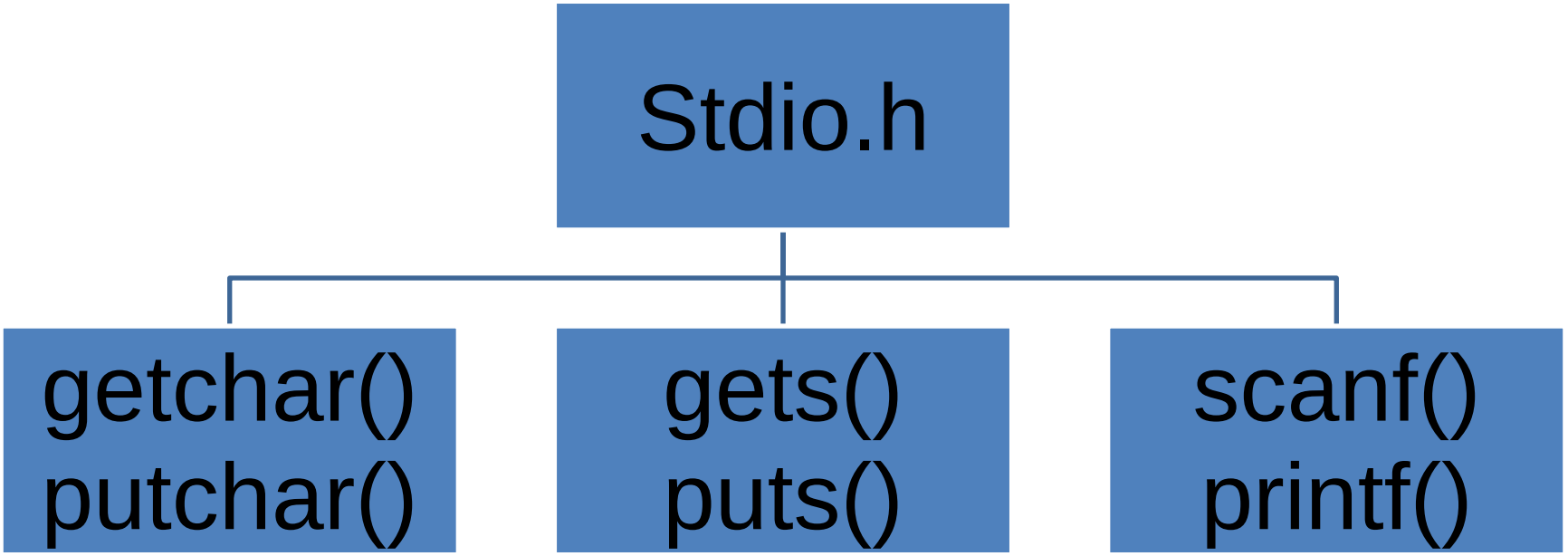
Multiline Comments in C

- Multi line comment can be placed anywhere
- Multi line comment starts with /*
- Multi line comment ends with */
- Any symbols written between '/*' and '*/' are ignored by Compiler

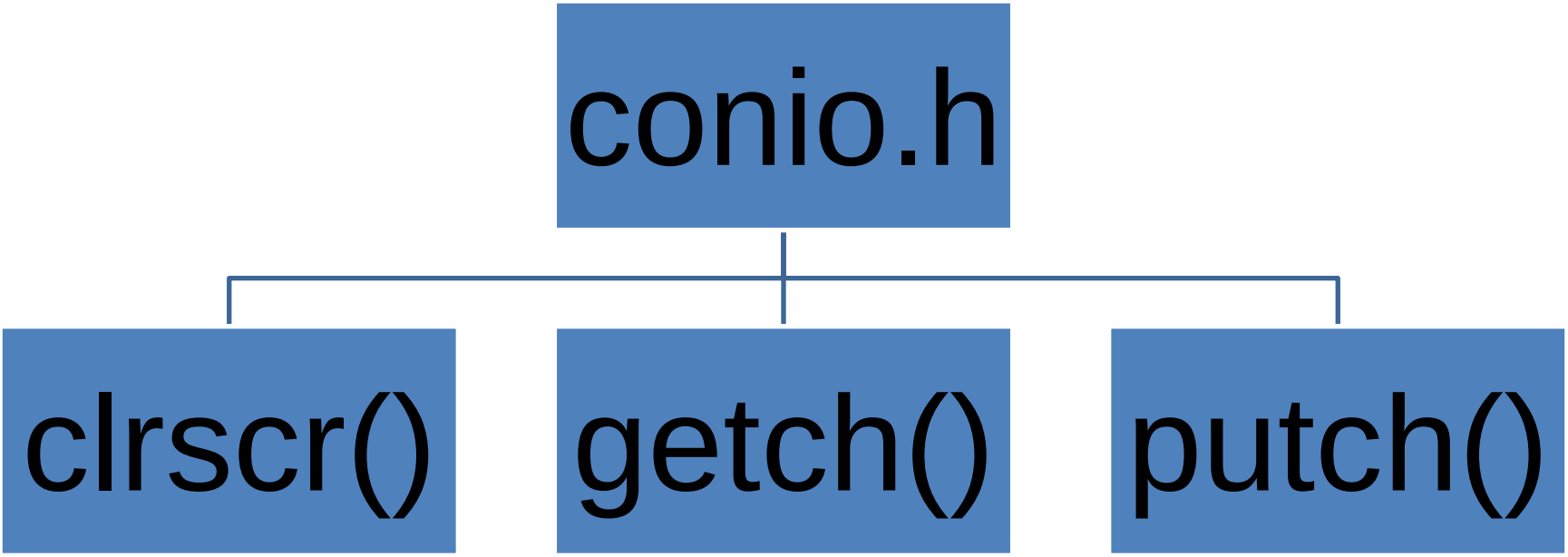
Character set of C

Alphabets	A, B, ….., Y, Z a, b, ……, y, z
Digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Special symbols	~ ‘ ! @ # % ^ & * ( ) _ - + = | \ { } [ ] : ; " ' < > , . ? /

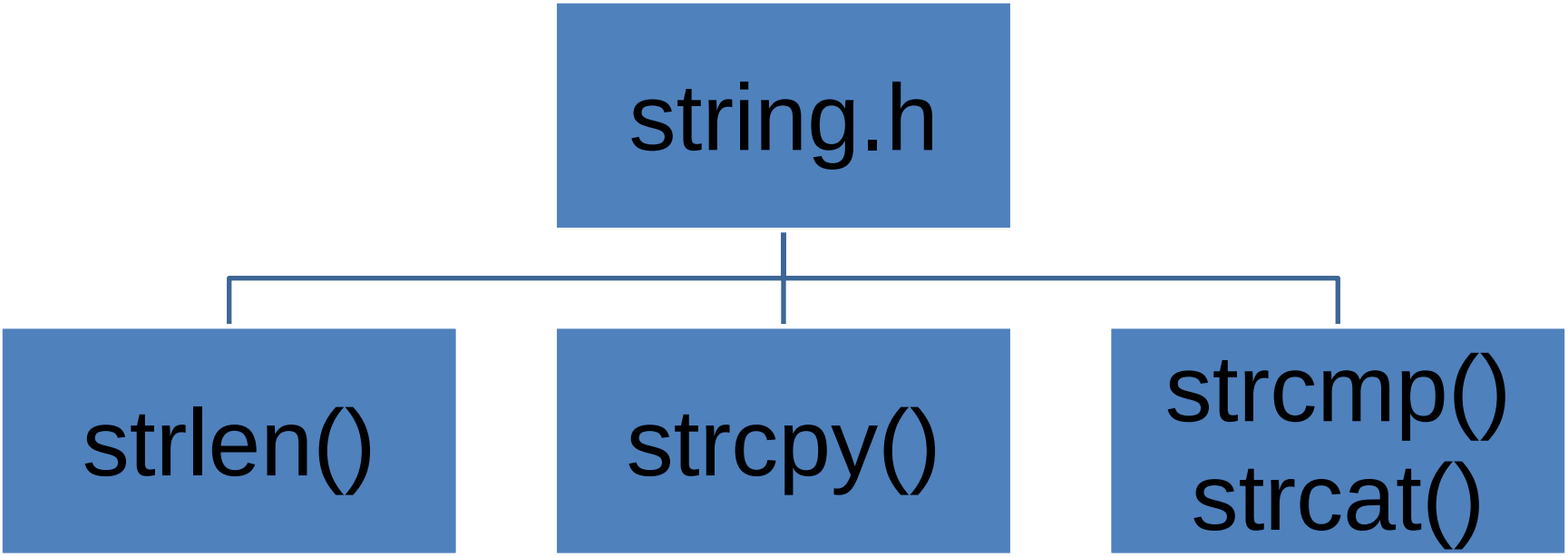
Preprocessor Directives (Examples)



Preprocessor Directives (Examples)



Preprocessor Directives (Examples)



Constants, Variables and Keywords

• alphabets, numbers and special symbols when properly combined form constants, variables and keywords

• Constant - entity that doesn’t change

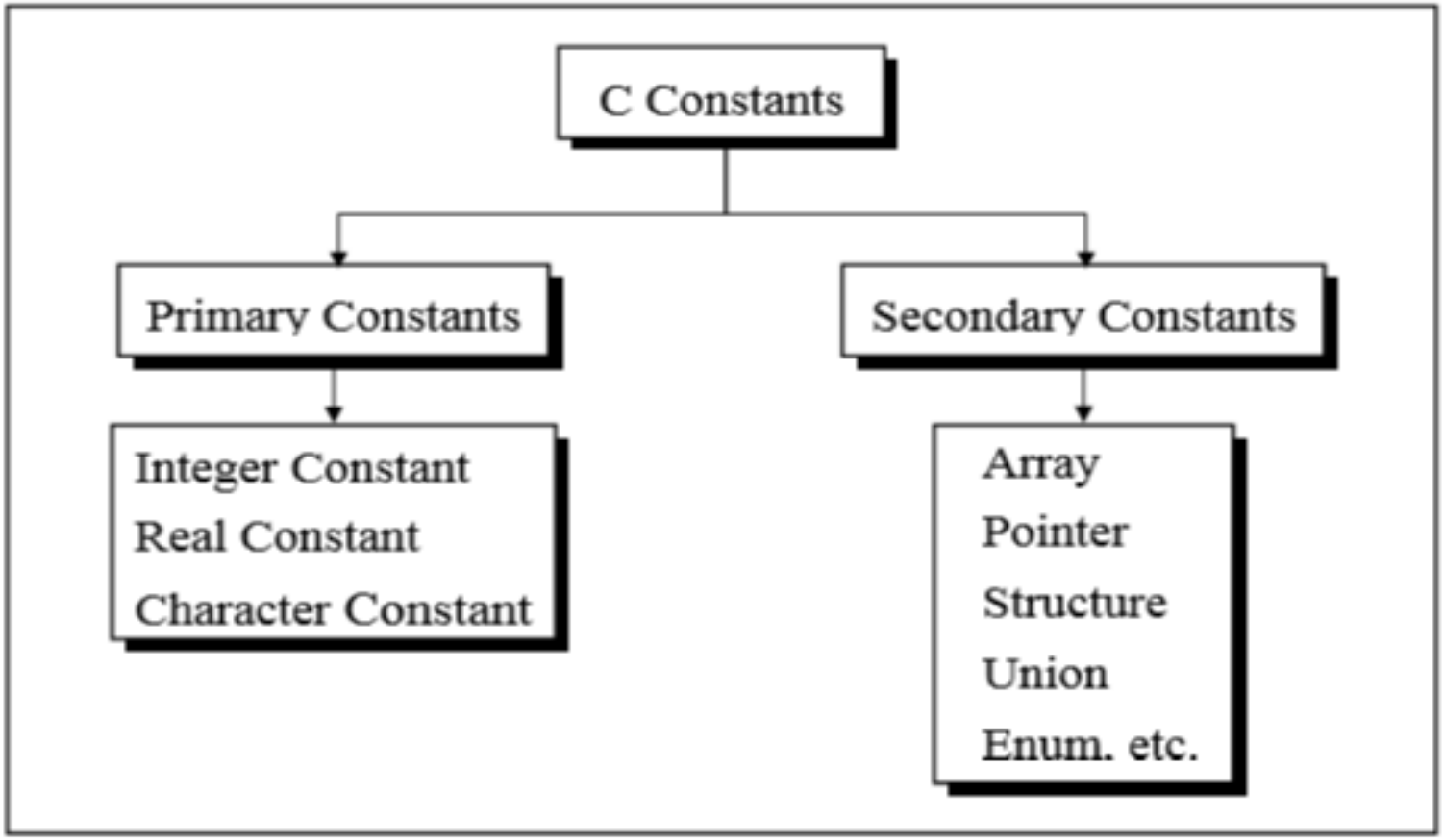
• Variable - entity that may change

• Keyword – reserved in programming language (Equivalent to words in natural language with predefined meaning)

Rules for Constructing Integer Constants

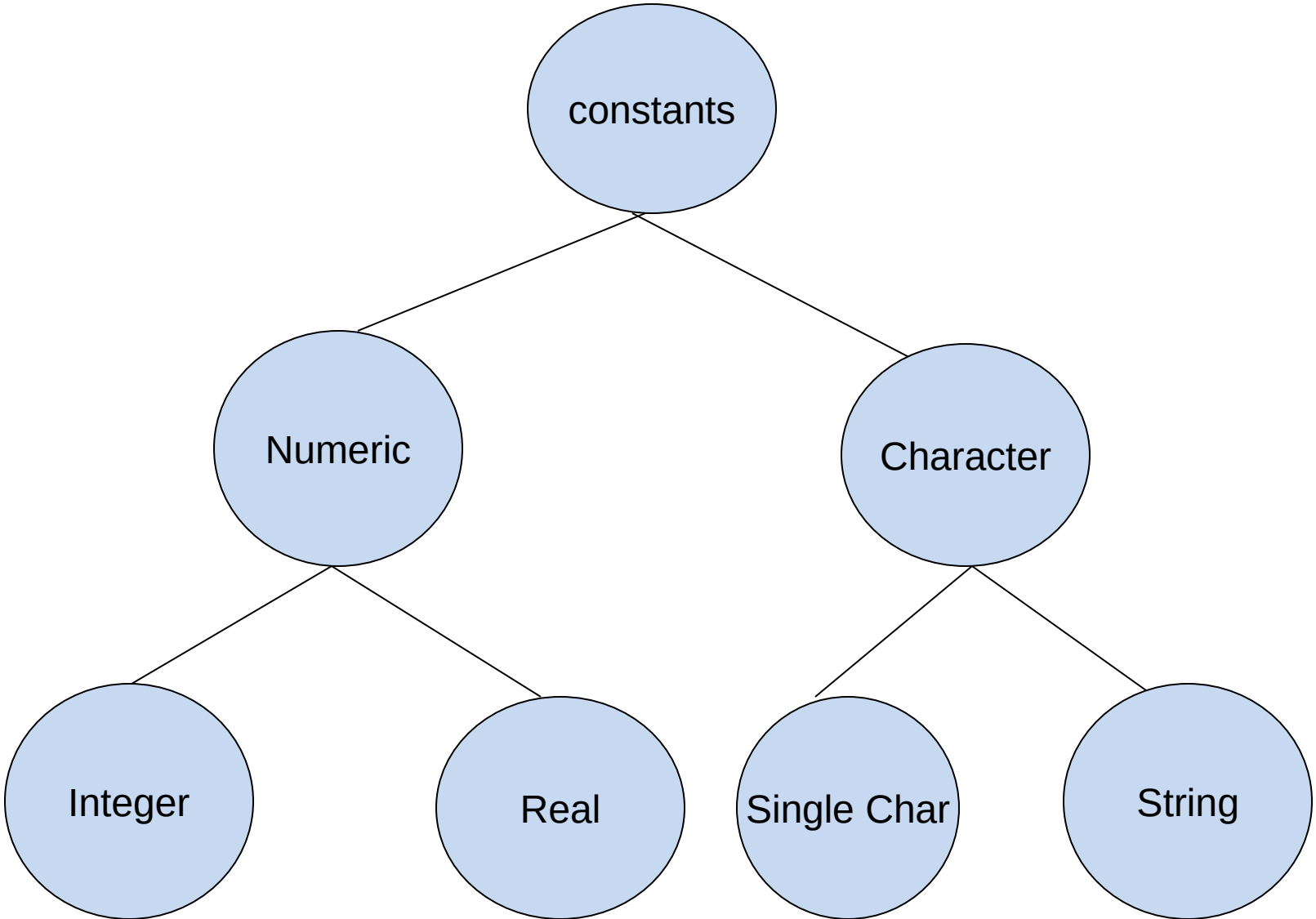
- Must have at least one digit
- Must not have a decimal point
- Can be either positive or negative
- If no sign precedes then it is assumed to be +ve
- No commas or blanks are allowed
- GCC is a 32-bit compiler therefore 4 bytes are allocated
- Range of values -2,147,483,648 to 2,147,483,647

Types of C Constants



Now Restrict Discussion to Primary Constants

Primary Constants



Constants – contd…

Numeric Constants:

- consist of numerals, an optional sign & an optional period.

Further classified into 2 types

Integer Constants

Real Constants

Constants – contd…

Integer Constants:

Integer constants are nothing but a whole number, without a fractional part.

Further classified into 3 types:

Decimal integer constants

Octal integer constants

Hexadecimal integer constants

Constants – contd..

Decimal Integer Constants:

The properties of Decimal integer constants are as follows:

It is a sequence of one or more digits (0-9)

It may have optional + or – sign, in absence of sign, the constant assumed to be positive

Constants – contd…

It should not have a period

Commas and blank spaces are not permitted

Example:

Valid decimal integer constants:

345 -987

Constants – contd..

Real Constants:

Also called as floating point constants

Are written in 2 forms:

Fractional form &

Exponential form

Constants – contd..

Fractional Form:

It must have atleast one digit & a decimal point

An optional sign (+ or -) can precede a real constant, in absence of sign, the value assumed to be positive

Commas or blank space are not permitted

Constants – contd..

Example:

Valid real constants:

345.67 -987.87

Invalid real constants:

3 45	Blank space not allowed
3,34	Commas not allowed
123	Decimal point missing

Constants – contd..

Exponential Form:

Offers a convenient way for writing very large and small real constants.

Example:

23000000.00, which can be written as 0.23E8 or 0.23e8 in exponential form.

0.000000123, which can be written as 0.123E-6 or 0.123e-6 in exponential form.

The letter E or e stands for exponential form.

Constants – contd..

Character Constants:

Consists of one or more characters of the alphabet of C.

Based on the number of characters present, it is further classified as:

Single character constant

String constant

Constants – contd..

Single Character Constant:

It consists of only one character and it is enclosed within a pair of single quotes.

Example:

‘a’ is a single character constant

‘z’ is a single character constant

Constants – contd..

String Constant:

It consists of one or more characters and is enclosed within pair of double quotes.

Example:

“abc” is a string constant

“Hello” is a string constant

Keywords & Identifiers

Keywords:

Keywords are those words of “C” which have predefined meaning assigned by the “C” language.

Should not be used for any purpose other than for which they are meant.

Keywords are also called “Reserved words”.

Keywords & Identifiers – cond…

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

Keywords & Identifiers – Cond..

Identifier:

- Identifier is the name which we supply in the program for variables, array names, function names etc.

Rules for valid Identifiers:

-can have max of 31 characters

-allowed characters in an identifier includes letters, digits and _

-should start with a letter

-Special character other than _ are not allowed

-should not be a keyword

Keywords & Identifiers – contd..

Examples:

Valid Identifiers:

number a1 a_1 a1b

Invalid Identifiers:

1a - starts with number

a 1 - Space is not allowed

Variables

A variable is a named memory location, the value of which can change during the execution of the program.

Variable name should be a valid C identifier

Variable name should reflects the purpose of the variable for better readability

Variables – contd..

Syntax:

data-type variable;

Note:

The declaration should end with a semicolon.

Example:

int a;

A memory location to hold an integer value gets allocated with the name.

Variables – contd..

To declare more than one variable of same type, the variables are seperated by commas as shown below.

Syntax:

Data-type variable1,variable2,…variablen;

Example:

int a,b,c;

Data Types

Data types refer to the classes of data that can be manipulated by C program.

Three fundamental Data types are:

- Character
- Integer
- Real

Data Types – contd..

int:

All numeric data items with no fractional part belong to integer type.

The size of variable is 2 bytes (in a 16 bit machine)

The range of values are -32768 to 32767

Data Types – contd..

float:

All numeric data items with fractional part belong to real or float type

Keyword “float” is used to declare the variables

A variable of “float” type required 4 bytes

The range of value that can be stored in it is 3.4e-38 to 3.4e+38.

Example:

```
float f;
```

Data Types – contd..

double:

A variable of double type requires 8 bytes of memory space

The range of values that can be stored in it is 1.7e-308 to 1.7e+308

Example:

```
double d;
```

Data Types – contd..

Size and range of values of different data types on a 16 bit machine

Data Types	Size (in bytes)
char	1
unsigned char	1
int	2
unsigned int	2
short int	2
unsigned short int	2
long int	4
unsigned long int	4
float	4
double	8
long double	10

Data Types – contd..

Size and range of values of different data types on a 32 bit machine

Data Types	Size (in bytes)
char	1
unsigned char	1
int	4
unsigned int	4
short int	2
unsigned short int	2
long int	4
unsigned long int	4
float	4
double	8
long double	10

Integer Types

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	4 bytes	-2,147,483,648 to 2,147,483,647
unsigned long	4 bytes	0 to 4,294,967,295

Floating Point Types

Type	Storage size	Value range	Precision
float	4 byte	1.2E-38 to 3.4E+38	6 decimal places
double	8 byte	2.3E-308 to 1.7E+308	15 decimal places
long double	10 byte	3.4E-4932 to 1.1E+4932	19 decimal places

Discuss Valid and Invalid variable names

- BASICSALARY
- _basic
- basic-hra
- #MEAN
- group.
- 422
- population in 2006
- FLOAT
- hELLO

Input-Output Operations

Accepting the required inputs from input devices and displaying the produced results on output devices are referred to as Input-Output operations.

The getchar() and putchar() Functions:

getchar() and putchar() are the simplest I/O functions and they are used to perform character input and output respectively.

getchar():

Is a function used to read a character through standard input device, keyboard.

Syntax:

```
c = getchar();
```

Where,

c is a character variable

Input and Output Operations

putchar():

Is used to display a character on standard output, monitor screen.

Example:

```
putchar(c);
```

Input-Output Operations – contd..

Sample Program for getchar() and putchar():

```
#include<stdio.h>

int main(void)
{
	char c;
	c = getchar();
	putchar(c);

	return 0;
}
```

Output:

```
r
r
```

Input-Output Operations – contd..

scanf() and printf() Functions:

scanf() function is used to accept mixed or same type of data through a standard input device, keyboard.

Syntax:

scanf(“control-strings”, arg1,arg2…argn);

Input-Output Operations – contd..

Example:

scanf(“%d”, &n);

Accepts a value into a variable ‘n’ of type int.

where,

%d is the format specifier for the int variable ‘n’.

Input-Output Operations – contd..

Format specifiers for other data types are given below:

Format specifiers	Meaning
%c	A character
%d	A decimal integer
%f	A floating point value
%hu	A short integer
%ld	A long integer
%s	A string
%u	An unsigned integer
%o	An octal number
%x	A hexadecimal number

Table of escape sequence:

Escape Sequence	Meaning
\n	New Line
\t	Horizontal Tab
\v	Vertical Tab
\f	Form feed
\b	Backspace
\a	Alert character
\\	Backslash
\?	Question Mark
\’	Single Quote
\”	Double Quote

Input-Output Operations – contd..

printf():

printf() is used to display data on the screen

Syntax:

printf(“control strings”, arg1,arg2…argn);

Example:

printf(“%d”, n);

To display the value of variable ‘n’ of type int.

Input-Output Operations – contd..

Formatting of Outputs:

Formatting of output refers to displaying the outputs in a more readable manner.

The main objective is to increase the degree of readability of outputs. It is accomplished with the specification of the width in which data items are to be displayed.

Input-Output Operations – contd..

Formatting of Integers:

To display the value of i of type int,

printf(“%d”, i);

The result would be the value of ‘i’ will be displayed on the screen starting from the first column of the screen.

3456

The value of i can be formatted by modifying the above control string. The modified control string would be %wd, w is an integer specifying the width with in which the value of i has to be displayed.

Sample C Program

```
/* To find out the area and circumference of Circle  */
#include<stdio.h>
#define PI 3.14

int main( )
{
        int r;
        float area, circumference;
        printf("Enter radius\n");
        scanf("%d", &r);
        area = PI * r * r;
        circumference = 2 * PI * r;
        printf("area=%f \n", area);
        printf("Circumference=%f \n", circumference);
        return 0;
}
```

Practice Problems

- Find the area and circumference of a circle
- Addition of two numbers
- Print the natural numbers 1-20
- Print the year is leap or not