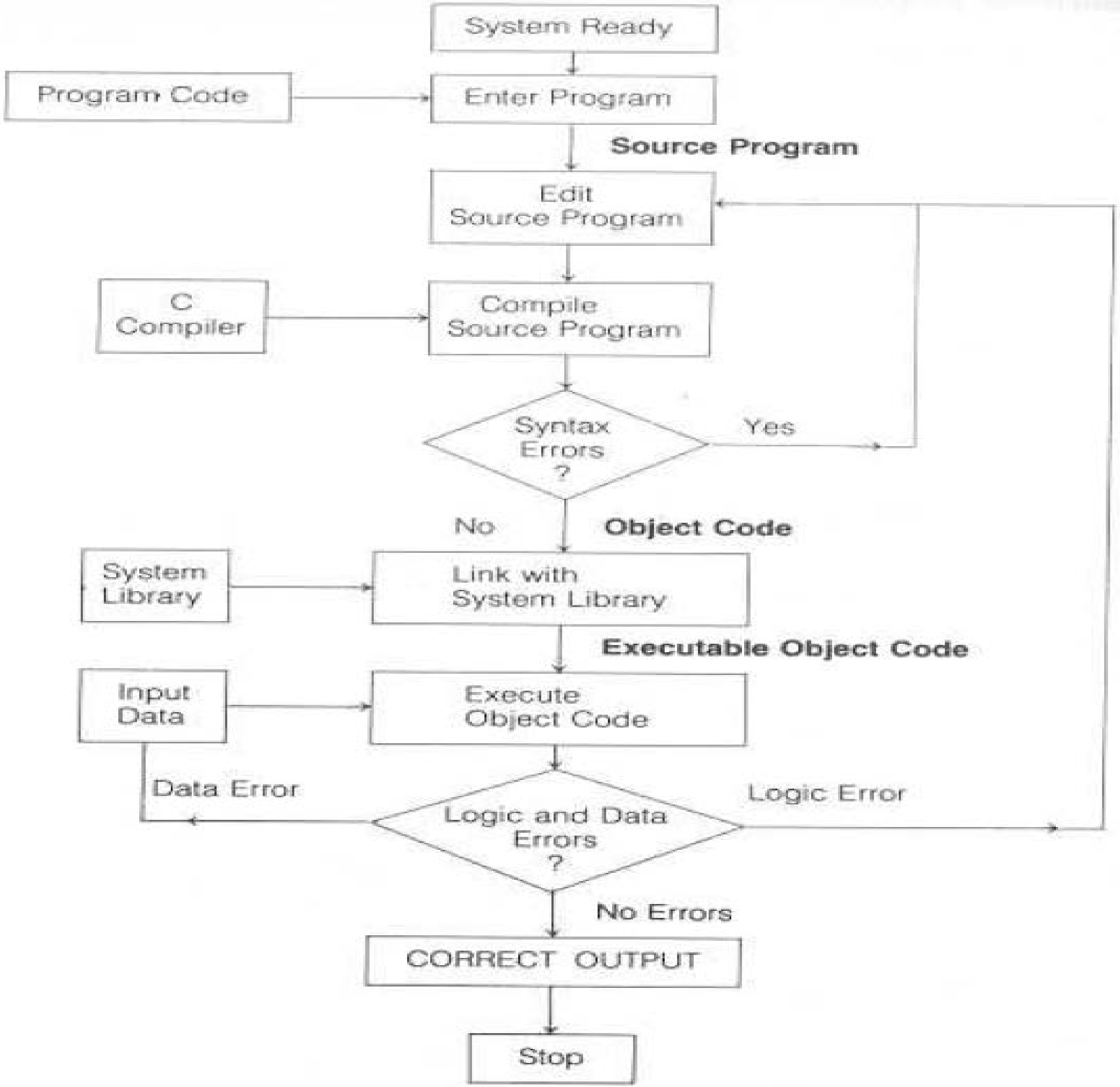
Elements of C

Contents

- How C program gets executed
- C Tokens
- Variables
- Primary Data Types

How the program gets executed

- Enter the program using any editor
- Edit the source program
- Compile the code
- If syntax error exists then edit the program again
- If syntax error does not exist then Object code is created (filename.obj)
- Execute the Object code and check input data
- The code check for errors
- If Logic error exists then edit the program
- If Data error exists input data is again given to the program
- If code is error free then output appears on screen


System Ready

Program Code

Enter Program

Source Program

Edit Source Program

C Compiler

Compile Source Program

Syntax Errors ?

Yes

No

Object Code

System Library

Link with System Library

Executable Object Code

Input Data

Execute Object Code

Data Error

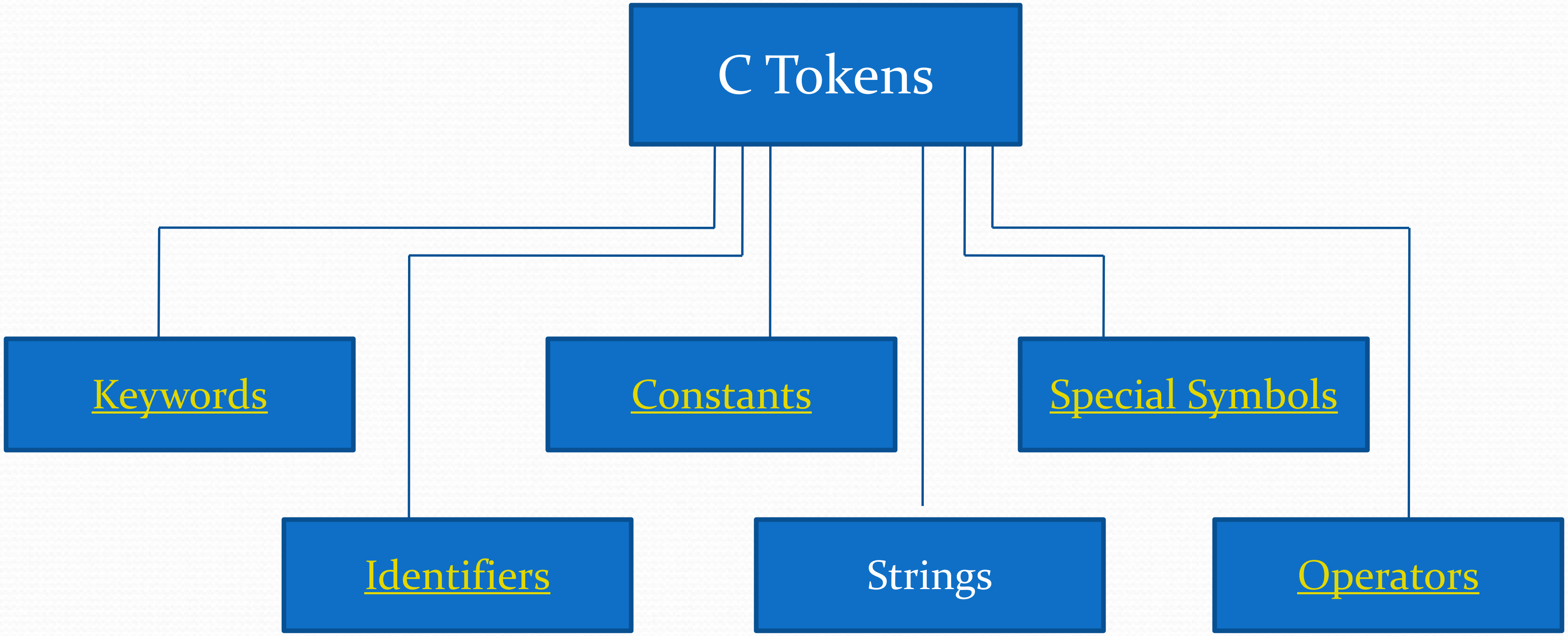
Logic and Data Errors ?

Logic Error

No Errors

CORRECT OUTPUT

Stop



C Tokens

Keywords

Constants

Special Symbols

Identifiers

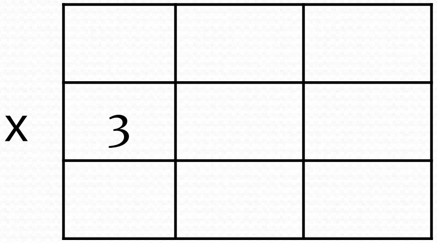
Strings

Operators

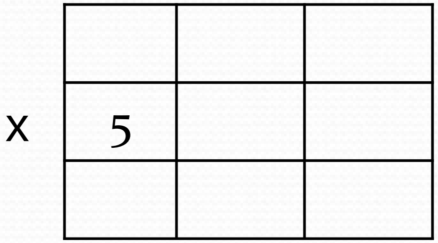
- In a passage individual words and punctuation marks are called as tokens
- In C individual units are called *Tokens*

Variables

- Variable is a name used to store values
- Values assigned to variable can be changed during execution of a program



x=3



x=5

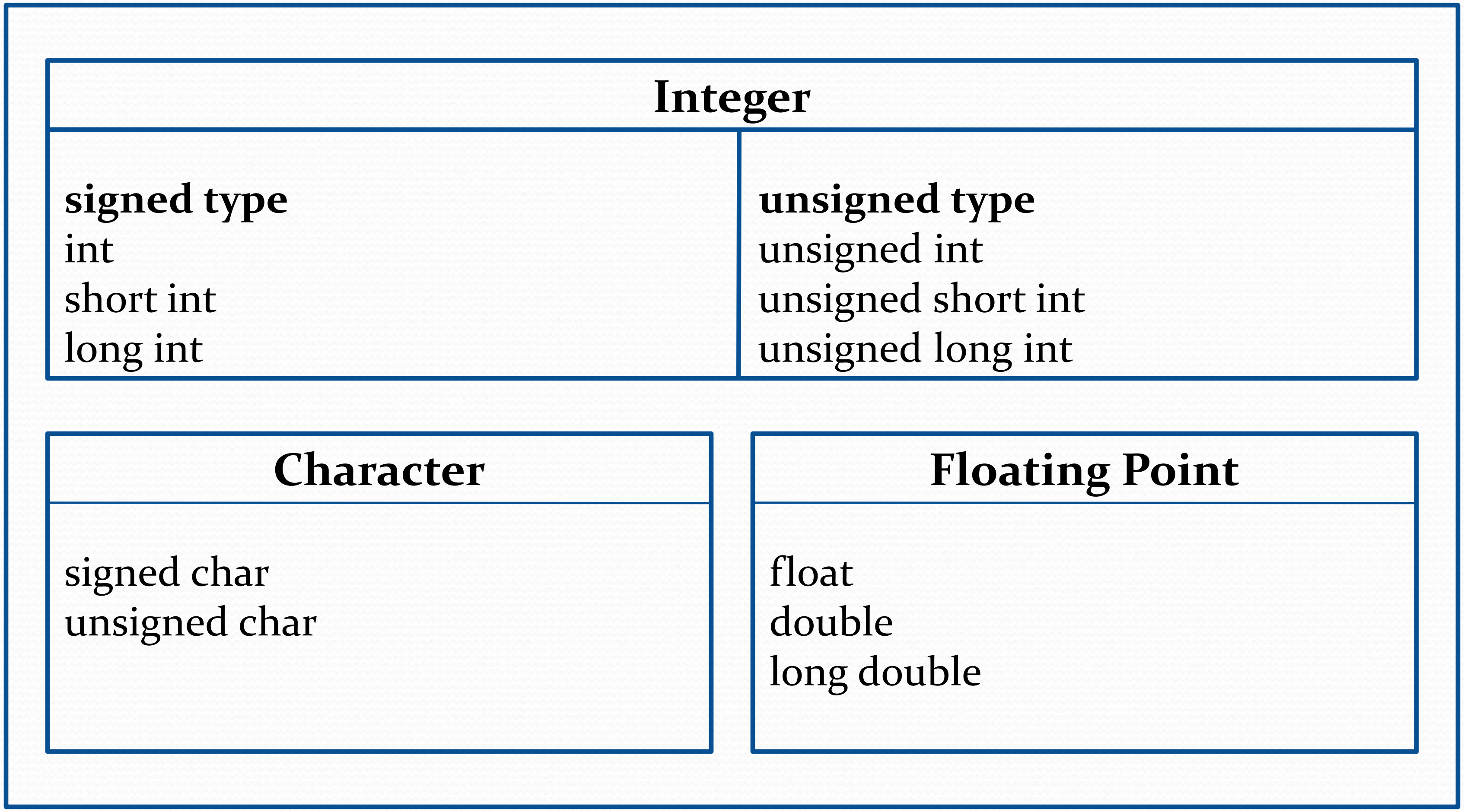
Data Types

- A type is a collection of values with shared properties
- Using types makes a program easier to read and understand
- Using types makes it easier for the compiler
- Types makes it easier to detect certain programming errors

Classes of Data Types

- Primary (Fundamental) Data Types
- User-defined Data Types
- Derived data types

Primary Data Types



Integer

signed type
int
short int
long int

unsigned type
unsigned int
unsigned short int
unsigned long int

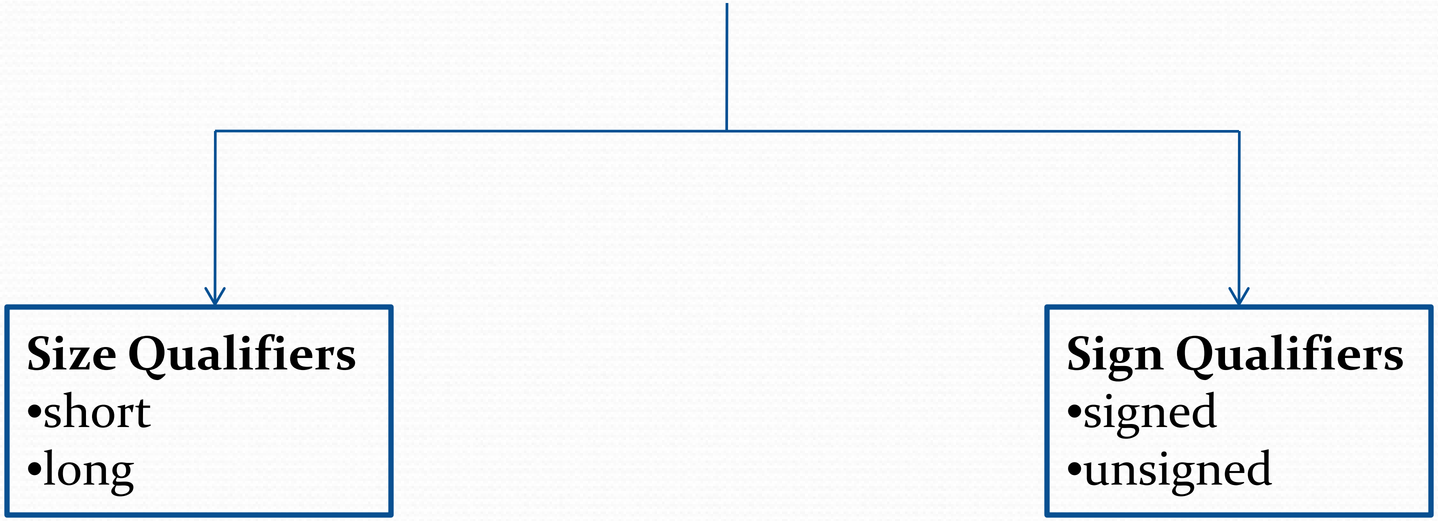
Character

signed char
unsigned char

Floating Point

float
double
long double

Use of Qualifiers



Size Qualifiers

- short
- long

Sign Qualifiers

- signed
- unsigned

Note:

❑If unsigned qualifier is assigned the number is always positive

❑If signed qualifier the number may be positive or negative

Range of Data Types

Basic Data Type	Data Type with qualifiers	Size (Bytes)	Range
char	char or signed char unsigned char	1 1	-128 to 127 0 to 255
int	int or signed int unsigned int short int or signed short int unsigned short int long int or signed long int unsigned long int	2 2 1 1 4 4	-32768 to 32767 0 to 65535 -128 to 127 0 to 255
float	float	4	3.4e-38 to 3.4e+38
double	double long double	8 10	1.7e-308 to 1.7e+308 3.4e-4932 to 1.1e+4932

Example

```
#include<stdio.h>
main()
{
   int a, b, c, d;           //declaration
   unsigned u = 10;          //declaration

   a = 12; b = -24;          //assignment
   c = a + u; d = b + u;     //assignment
   printf("a+u=%d, b+u=%d\n", c, d);
}
```

format specifiers

output:

```
a+u=22, b+u=-14
```

- -2^{n-1}

  to

  $+2^{n-1} - 1$