

Southern Technical University
Technical Institute / Qurna
Dep. of Computer Systems Techniques

Second class

Subject : Data Structures

Lecturer : Israa Mahmood Hayder

Lecture no.3

اسلوب تمثيل المياكل البسيطة
(Primitive Data Structures)

- الاسبوع الثالث -

B// Rationale (مبررات الوحدة) :-

The primitive data structure is the simplest and very important type to start with and to know how they are represented, how much memory size required to represent each of them, and the functions that used with these types in C++ language.

C// Central Ideas (الفكرة المركزية):-

- Strings
- Logical Data
- Pointers

D// Objectives (أهداف الوحدة):-

After studying this unit , the student will be able to:-

- Represent logical data in memory
- Represent characters using ASCII and EBCDIC Methods.
- String manipulation using Markov algorithm.
- Using addresses and pointers in C++

Circle the correct answer:

- 1- To represent the logical data in memory we require:
a) 8bit b) 1 byte c) 1bit
- 2- the statement `a= 4>7` in C++ is:
a)False b>true c)error statement
- 3- `a= 3>0 && 4%d=0` statement in C++ is:
a)False b>true c)error statement
- 4- **atol()** : convert string to :
a)integer b)long c)character
- 5- **isdigit()** : if string is digit gives:
a) No>0 b)No>=0 c) No< > 0
- 6- When `k=8`, and `p` is pointer to `k` then:
a) `*p= 8` b)`*p= 0x3fffd1c` c) `p=8`
- 7 - Which of the following statements is false:
a) `int *p='66'` b)`long *p=&n` c)`int *p=0x3fffd1b;`
- 8- **islower()** : if string is lower case gives
a) No<>0 b)No>=0 c) No< 0
- 9- `bool a=0.0` is:
a)True b)False c)error statement
- 10- In ASCII groups:-Bit 6& Bit 7=01 means that the char is:
a) Upper Case Char b) Special Char c) Digits and Punctuation

A// Methods for representing Characters:

A *character* abstract data type is represented as an integer value that corresponds to a character set. A *character set* assigns an integer value to each character, punctuation, and symbol used in a language.

For example, the letter *A* is stored in memory as the value 65, which corresponds to the letter *A* in a character set. The computer knows to treat the value 65 as the letter *A* rather than the number 65 because memory was reserved using the `char` abstract data type. The keyword `char` tells the computer that the integer stored in that memory location is treated as a character and not a number.

There are two character sets used in programming, the American Standard Code for Information Interchange (ASCII) and Unicode. ASCII is the granddaddy of character sets and uses a byte to represent a maximum of 256 characters. However, a serious problem was evident after years of using ASCII. Many languages such as Russian, Arabic, Japanese, and Chinese have more than 256 characters in their language. A new character set called Unicode was developed to resolve this problem. Unicode uses 2 bytes to represent each character. Choose a `char` whenever you need to store a single character in memory.

Boolean Abstract Data Type

A Boolean abstract data type reserves memory to store a Boolean value, which is a true or false represented as a zero or one. Choose a Boolean whenever you need to store one of two possibilities in memory.

Logical Data : It takes True or False stored as 1 bit or more (1 for true , 0 for false) or (11111111 for True and 00000000 for False) or half byte (like Pascal) C++ need 1 bit to represent Boolean value.

`bool a,b`

`bool a='a'` `True`

`bool a=0.0` `False`

`a= 5>7` `False`

`b= 7==8-1` `True`

`a= 3>0 && 4%d==0`

Relational operators

`!` not

`&&` logical and


`| |` logical OR

Characters :- most widely used character set are represented by :-

- 1) ASCII (American standard code for info interchange)
- 2) EBCDIC (Extended Binary Code for decimal ")

Ex

	B ₇	B ₆	B ₅				B ₀
E	0	1	0	0	0	1	0
e	0	1	1	0	0	1	0



ASCII groups:-	Bit 6	Bit 5	
	0	0	Control Char
	0	1	Digits and Punctuation
	1	0	Upper Case Char and Special Char
	1	1	Lower Case

String :- I s a sequence of characters or string functions with characters and strings in C++ .

- 1) **atol()** : convert string to long

Ex long atol (const char*s);

or

```
long x;
```

```
char * s ;
```

```
cin << s
```

```
cout >> " s= " >> atol (s);
```

- 2) **atoi()** : convert string to integer ;

```
int atoi(const char * s);
```

- 3) **atof()**: convert string to double

```
double atof(const char * s);
```

- 4) **isalnum()** : if string is number gives No< > 0
- 5) **isalpha()** : if string is alphabetic gives No< > 0
- 6) **isdigit()** : if string is digit gives No< > 0
- 7) **islower()** : if string is lower case gives No< > 0
- 8) **isupper()** : if string is upper case gives No< > 0
- 9) **strlen()** : find length of string
- 10) **strcat()** : concatenate 2 strings
- 11) **strcmp()** : compare 2 strings
- 12) **strcpy()** : copy string to another (no of char must be defined)
- 13) **strstr()** : find first occurrence of substring in other string.

User defined data types

Contain any combination of primitive data structures the following structure defines a student record :-

Struct stud

```
{  
    int  studname;  
    char grade;  
};
```

Ex Program

```
#include <iostream>  
using namespace std;  
Struct stud  
{  
    int studname;  
    char grad;  
}  
Void main()  
{
```

```

stud mystud;
mystud.grad=10;
mystud.studname='A';
    cout <<"grad:" << mystud.grad << " " << mystud.studname << endl;
};

```

Define a class : structures used with classes to group primitive a class definition of a structure in C++ as follows :-

```

Class stud {
    int sname;
    char grade;
    void displ( ) {
        cout << "student:" << st << "grade:" << grade << endl;
    }
};

```

To Access members of a class:

```

Stud.sname= 'AA';
Stud.grade( ) ;

```

Pointers :

Is dynamic variable is created and destroyed dynamically during the execution of the program, unlike static variables.

Dynamic variables are not referenced by user specified name , instead , they are referenced by pointers.

P → V

The dynamic variable "V" is referenced by pointer variable "P" which "points to" V.

int n; (static variable)

ox3ffdl4
33

n
int

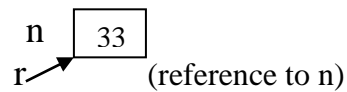
cout << n ; we use name to print the value. This expr. Print the value n(33).

cout << & n ; print the address of n (ox3fffd14)

Using references :-

Ex1

```
Main()  
{  
    int n=33 ;  
    int & r=n ;  
    cout<< "n=" << n << ",r=" << r << endl;  
    -- n;  
    cout << "n=" << n << ",r=" << r << endl ;  
    cout << "&n=" << & n << ",& r=" << &r<< endl ;  
}
```



عنوان n

Run

n=33 ,	r=33
n=32 ,	r=32
ox3fffd13	ox3fffd13

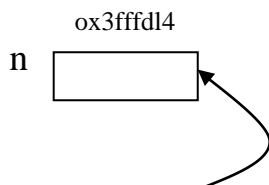
The address can be saved in variable called pointer like int* that points to integer.

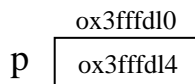
Ex2

```
Main()  
{  
    int n=33 ;  
    int *p=&n ;  
    // p hold the address of n  
  
    cout<< "n=" << n << ",&n" <<&n<< ",p=" << p << endl;  
    cout << "&p=" << &p << endl ;  
}
```

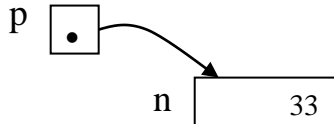
Run

n=33 ,	&n= ox3fffd14 ,	p= ox3fffd14
&p= ox3fffd10		





and the last pointers can be drawn as follows :-



p points to n then *p = 33 like n

Ex3

```
main ( )
{
    int n = 33;
    int *p = &n;
    cout << "**p" << *p << endl ;    }
```

Run

*p=33

Quiz1:

Write program to define pointer to long integer and decrement by 1 then print the pointer value and address.

String Manipulation (Markov Algorithm)

It is a formal system for string manipulation that was originally developed to support a theory of computation. In markov algorithm general strategy is to take x as input string and through number of steps (or productions) in the algorithm, transform x to an output string y .


Ex

Consider the markov algorithm having the following productions :-

- P1 : 'ab' → 'b'
- P2 : 'ac' → 'c'
- P3 : 'aa' → 'a'

Execute it on the given string 'bacaabaa' solution :

bacaabaa ⇒ bcaabaa (by P2)
 bcaaabaa ⇒ bcabaa (by P3)

bcabaa \Rightarrow bcbaa (by P1)
 bcbaa \Rightarrow bcba (by P3)
 output string 

Quiz2:

Consider the Markov algorithm having the following productions, and execute it on the input string ' xxxxyxxxxzxxx ':

P1: ' xxx ' \rightarrow 'x'

P2: ' xxxy ' \rightarrow 'y'

P3: ' yxxx ' \rightarrow 'x'

P4: ' xxz ' \rightarrow 'z'

A// Circle the correct answer:

- 1- In C++ the relation (7==6+1) gives:
 - a) true b) false c)error
- 2- The function isupper(x) gives 0 when
 - a) x='A' b) x='a' c)x='4'
- 3- When n=5, and p is pointer to n then:
 - a) *p=0x3fffd1a b)*p=5 c) p=5
- 4- Which of the following statements is false:
 - a) char *p='a' b)int *p=&n c)int *p=0x3fffd1b;
- 5- **atof()**: convert string to:
 - a)integer b)long c)real
- 6- b= 7==8-1 statement in C++ is:
 - a)True b>false c)error
- 7- If P is pointer P++ means:
 - a)increment address of pointer b)increment value that pointer point
- 8- if b=33 then a= b==32-1 statement in C++ is:
 - a)False b>true c)error statement
- 9- The ascii of a char is (0 1 1 0 0 1 0 1) means that the char is:

a) upper case letter b) control char c) small case letter

10- The statement : `not(5+2=10)` in C++ is:

a) False b) true c) error statement

References:

1- <http://en.kioskea.net/s/representation-of-real-numbers-and-integers>

2 - سلسلة ملخصات شوم للبرمجة بلغة C++، الجزء الاول، الطبعة الاولى، جون ر. هيوبارد، الدار الدولية للاستثمارات الثقافية، ٢٠٠٠.

٣- هياكل البيانات / الطبعة الثانية، تأليف د. عصام الصفار، اصدارات السفير للنشر/ بغداد، ٢٠٠١.

٤- الحقيبة التعليمية مادة "هياكل البيانات"، اعداد : نفارت الياس يوسف، المعهد التقني كركوك.