



COP 3503 Lab#2

Loops and Arrays

Print Special characters

	Description
<code>\n</code>	Linefeed(new line)
<code>\t</code>	Tab (to align elements)
<code>\r</code>	Carriage return
<code>\b</code>	Backspace
<code>\\</code>	Backslash
<code>\'</code>	Single quote
<code>\''</code>	Double quote

Print Special Characters

- Examples :

1) `char c = '\n';`

```
std::cout<<"line one" <<c<< "line two"<<c;
```

Output:

line one

line two

2) `std::cout<<"tab\\1\ttab\\2";`

Output:

tab\1 tab\2

Loops(1): For Loop

```
for (initial-action; condition; increase) {  
    //loop body Statements(s);  
}
```

- ➤ The **initial-action** and **increase** are optional, which can be
- 1) ***empty*** or 2) ***a list comma-separated statements***
for (int i=0, j=8; (i+j<10); i+=2, j-=1) { //do something here }
- ➤ When the **condition** is empty, it is implicitly true

Loops(2): While and Do-while

Syntax:

```
while (condition)
{
// Loop body Statement(s);
}
```

```
do
{
// Loop body Statement(s);
} while (condition);
```

Loops(3): Tips

- You may use “**break;**” or “**continue;**” inside loops:
 - continue**: end current iteration and start with next one;
 - break**: jump out to execute code after the loop.
- Use the most intuitive loop :
 - If number of repetitions known ***for***
 - If number of repetitions unknown ***while***
 - And if should be touched at least once (before testing the condition) ***do-while***

Loops(4): examples

```
#include <iostream>
int main(){
    int n; std::cin >> n; int k = 0;
    for (int i = 1; i <= n; i++){
        std::cout<<i<< "\t";
        if (i % 10 == 0){ //10 numbers per line
            std::cout << "\n";
        } // end if
    } // end for loop
    std::cout << "\n";
    return 0 ;
}
```

Arrays(1): Declaration

type name [elements];

Type: such as: int, float, char...

Name: identifier

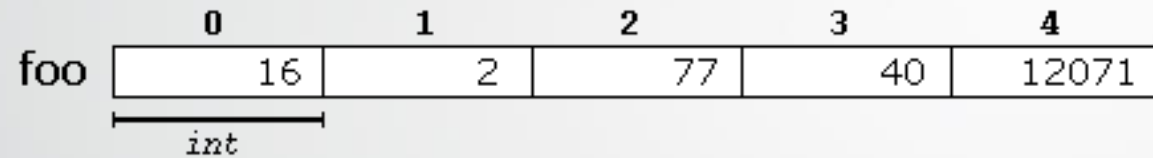
Elements: specifies the length of the array in terms of the number of elements.(must be a ***constant expression***)

Example:

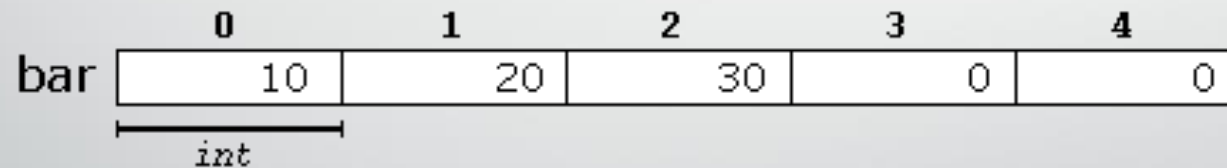
```
int foo [5];  
const int size = 6;  
char name [size];
```


Array(2):Initialization

```
int foo [5] = { 16, 2, 77, 40, 12071 };
```

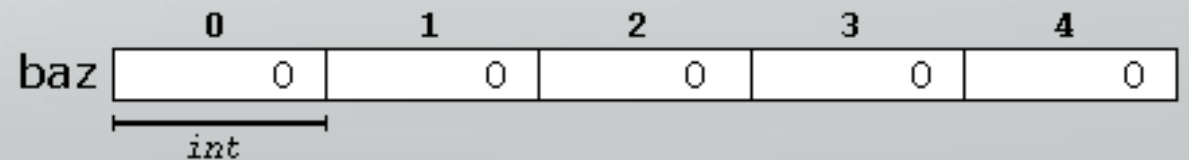


```
int bar [5] = { 10, 20, 30 };
```



So how about?

```
int baz[5] = {};
```



Array(3):Dynamic Array

- pointer = new type [**size**]

Dynamic Array don't need constant variable as size. Use **new** function to assign memory during runtime using any variable value as size.

```
int main (){  
    int size; cin>>size;  
    int * foo = new int [size]; //foo is a pointer to an array  
    for( int i = 0; i < size; i++ ) foo[i] = 0;  
    delete [ ] foo;  
    return 0;  
}
```

Multidimensional arrays

		0	1	2	3	4
jimmy {	0					
	1					
	2					

```
int jimmy [3][5];
```

Multidimensional arrays are just an abstraction for programmers, since the same results can be achieved with a simple array, by multiplying its indices.

```
int jimmy [3][5]; // is equivalent to  
int jimmy [15]; // (3 * 5 = 15)
```

Example

multidimensional array

```
#define WIDTH 5  
#define HEIGHT 3
```

```
int main () {  
    int jimmy [HEIGHT][WIDTH];  
    int n,m;  
  
    for (n=0; n<HEIGHT; n++)  
        for (m=0; m<WIDTH; m++) {  
            jimmy[n][m]=(n+1)*(m+1);  
        }  
}
```

pseudo-multidimensional array

```
#define WIDTH 5  
#define HEIGHT 3
```

```
int main () {  
    int jimmy [HEIGHT * WIDTH];  
    int n,m;  
  
    for (n=0; n<HEIGHT; n++)  
        for (m=0; m<WIDTH; m++) {  
            jimmy[n*WIDTH+m]=(n+1)*(m+1);  
        }  
}
```

Exercise 1:

- Print out the two dimensional matrix we produced in last example as following.

```
output: 1    2    3    4    5;  
        2    4    6    8    10;  
        3    6    9    12   15;
```

Exercise 2

Write a program that asks the user to input the UFID and store each digits into an array. The program then :

- 1) compute and print how many digits are greater than the first digit and
- 2) give the sum of those digits less than or equal to first digit.

Example results:

input: 48632136

output: 8,6,6 are greater than the first digit 4.

Sum of other digits is 9.

Exercise 3

- Write a program that asks the user to input 10 integers of an array and an integer value V. Search if the value V exists in the array and then remove the first occurrence of V, shifting each following element left and adding a zero at the end of the array.
- The program need to print the final array.

input: 10 integers: 2 3 4 5 6 5 4 3 2 1

Search for: 4

output: 2 3 5 6 5 4 3 2 1 0

- 
- Questions?