# Northern University, Nowshera

**Programming Fundamentals (CS-)**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776

**(Week 4) Lecture 7 and 8**

**Objectives:** **Learning objectives of this lecture are**

- **Unary Operator**
- **Char data type**
- **Arithmetic operator on char data type**
- **Integer vs char**

**Text Book & Resources:**

**Text Book:** Robert Lafore, "Object Oriented Programming", 4th Edition.

**Reference Book:** Greg Wiegand , "Teach Yourself C++ in 21 days", 2nd Edition. Online version.

# Northern University, Nowshera

**Programming Fundamentals (CS-)**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776

## Lecture # 7 Taking Input and Writing Program Unary Operators and Char Data type

In C++ cin statement is used to take input from user.

Syntax for cin is as follow.

***cin >> identifiername ;***

With cin statement identifier name is compulsory, that identifies the memory address where input value will be stored. Consider given piece of code.

```
int num;          ➔ At execution of this statement memory will be created

cin >> num ;      ➔ Will take input assume that user enters 21. This number is stored in num.

cout << "You have entered "<<num<<endl;  ➔ Will display Output
```

**Program:** Program actually is set of instruction to perform some specific task. For a programmer it is very necessary that he/she should know the behavior of each written, also he/she should be clear about the logics of program. For a beginner it's very important to understand what statement to be written and where is to be written. Let's practice it.

**Exercise:**

**Program Number 1**: Write a C++ program that takes two numbers as input, after taking input add both number and display the answer.

| Steps | C++ code |
|---|---|

# Northern University, Nowshera

**Programming Fundamentals (CS-)**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776

| | |
|---|---|
| 1) Take two numbers as input, as we discussed for an input it's necessary to have memory. First we will declare two identifiers.<br><br>2) Take input in both identifiers .<br>3) Add them.<br><br>4) Display answer. Is statement number 3 storing answer for display? Answer is ***NO.***<br><br>5) Modifying step 3 Add numbers and store it in some memory.<br>6) Display answer. | `#include "iostream"`<br>`using namespace std;`<br>`void main()`<br>`{`<br>`int num1, num2 ;`<br><br>`cout << "Enter number 1 ";`<br>`cin >> num1;`<br>`cout << "Enter number 2 ";`<br>`cin >> num2;`<br><br>`num1 + num2 ;`<br><br>`int answer = num1 + num2;`<br><br>`cout << "Sum = " << answer << endl;`<br><br>`}` |

**Program Number 2:** Suppose that you have to calculate bill of a product bought by a customer. You have to input quantity of a product purchased by user and price of 1 one product, after that display bill to be paid.

| Steps | C++ code |
|---|---|
| Above statement clearly states about taking input which things to be input i) quantity ii) price.<br>1) For input of price and declare two identifiers one for storing price and other for quantity.<br>2) Take input in both identifiers.<br>3) Declare third identifier for storing bill.<br>4) Multiply quantity and price and assign it to bill.<br>5) Display bill; | `#include "iostream"`<br>`using namespace std;`<br>`void main()`<br>`{`<br>`int quantity , unitprice ;`<br><br>`cout << "Enter quantity purchased ";`<br>`cin >> quantity;`<br>`cout << "Enter price of one product ";`<br>`cin >> unitprice;`<br><br>`int bill;`<br><br>`bill= quantity * unitprice ;`<br><br>`cout << "Bill = " << bill << endl;`<br><br>`}` |

**Program Number 3:** You are required to make an application for utility bill generator. You have to input units consumed from user along with cost per unit and display bill to be paid.

# Northern University, Nowshera

**Programming Fundamentals (CS-)**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776

| Steps | C++ code |
|---|---|
| Above program is similar to program number 2<br>1) Input unit consumed and price per unit. Before taking input declare two identifiers.<br><br>2) Multiply unit consumed and price per unit and assign it to third identifier and display it. | `#include "iostream"`<br>`using namespace std;`<br>`void main()`<br>`{`<br>`int unitconsumed , unitprice ;`<br>`cout << "Enter unit consumed ";`<br>`cin >> unitconsumed;`<br>`cout << "Enter price of one product ";`<br>`cin >> unitprice;`<br><br>`int bill=unitconsumed * unitprice ;`<br>`cout << "Bill = " << bill << endl;`<br><br>`}` |

**Program Number 4:** In norther university many students apply for BSCS program. You have to input their metric and inter percentage after that calculate average and display average percentage.

| Steps | Code |
|---|---|
| 1) As we have to input percentage of metric and inter we need to create two memories of float type.<br><br>2) Take input in both the memories.<br><br>3) Declare another memory for storing average percentage.<br><br>4) Compute Percentage and assign it to identifier declare in step 3.<br><br>5) Display value of identifier declared in step 3. | `#include "iostream"`<br>`using namespace std;`<br>`void main()`<br>`{`<br>`float metricper, interper;`<br><br>`cout << "Enter Metric Percentage ";`<br>`cin >> metricper;`<br>`cout << "Enter Inter Percentage ";`<br>`cin >> interper;`<br><br>`float avgper;`<br><br>`avgper = (metricper + interper) / 2;`<br><br>`cout << "Average Percentage : " << avgper;`<br>`}` |

# Northern University, Nowshera

**Programming Fundamentals (CS-)**

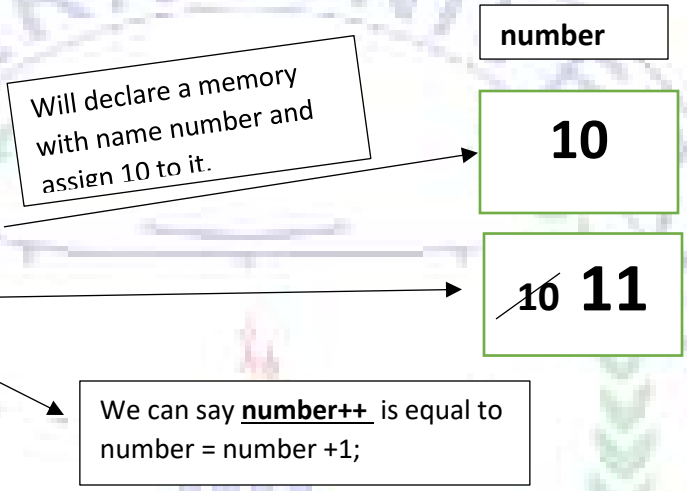
Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776

## Unary Operator:

Let's first look at two types of unary operators ++ (increment) and -- (decrement). ++ adds 1 in a memory and -- subtracts 1 from a memory. Unary operators requires only one operand. Both Increment and decrement are divided into further two categories prefix and postfix. When operator is used before some variable it is called prefix and when used after some variable it is called postfix. Further detail will be explained in programs given below.

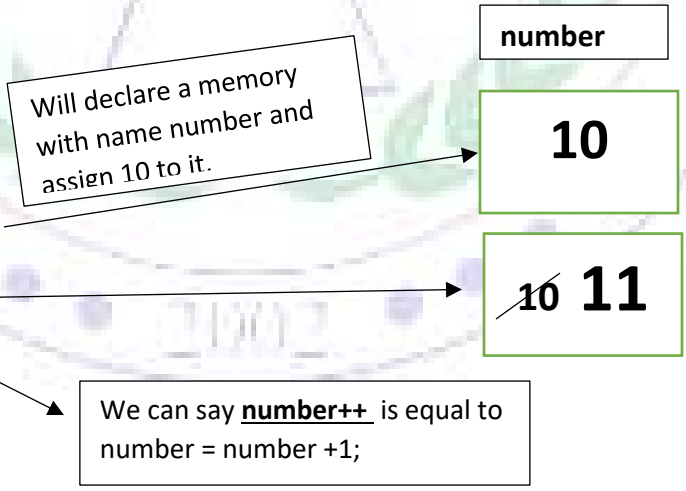
### Program 1:

```
#include "iostream"
using namespace std;
void main()
{
      int number = 10;

      number++;
}
```




### Program 2:

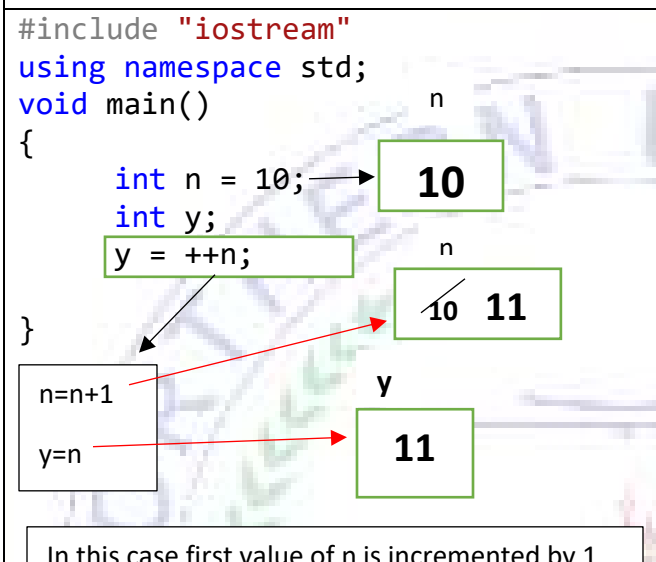
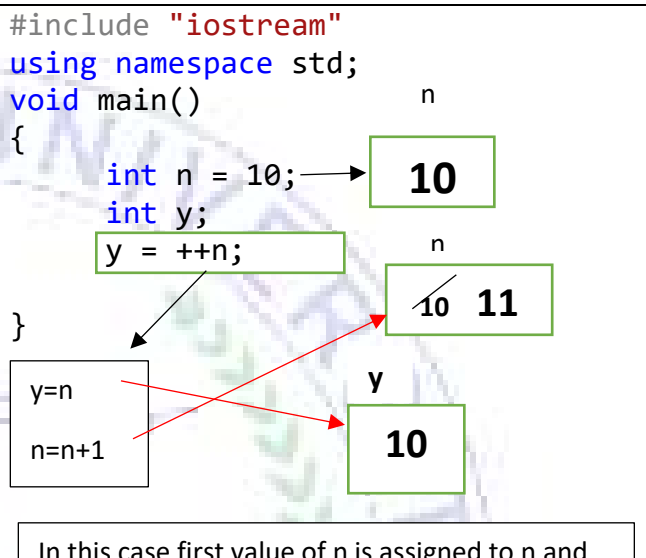
```
#include "iostream"
using namespace std;
void main()
{
      int number = 10;

      ++number;
}
```



# Northern University, Nowshera

**Programming Fundamentals (CS-)**
Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776

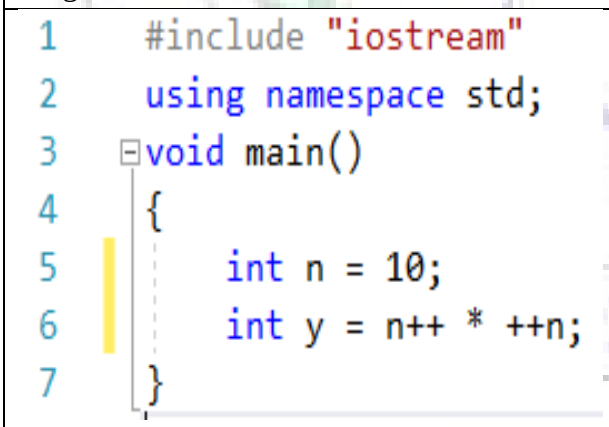
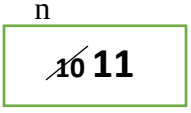
++ number ➔ Prefix Increment (Pre Increment) and number++ ➔ Postfix Increment (Pre Increment). So far from the above explain it seems that there is no difference between pre and post increment/decrement. Let us dig deep into it and find difference between pre and post.

| Pre Increment | Post Increment |
|---|---|
|  |  |
| In this case first value of n is incremented by 1 and then assigned to y. | In this case first value of n is assigned to n and then incremented |





Consider the program given.

| Program | Memory |
|---|---|
|  | At line number 5 a memory block is created with name n and inside that memory 10 is assigned.<br>  <br>Statement at line 6 is break down into four sub operations<br>**Operation 1 (pre increment)**<br>n=n+1 ( pre increment having highest priority)Memory looks like<br>  <br>**Operation 2 (Multiplication)** |

# Northern University, Nowshera

**Programming Fundamentals (CS-)**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776

| | |
|---|---|
| | n*n (multiplication is prior to post increment)<br>value inside memory n= 11 so 11*11➔121<br>**Operation 3 (Assignment to y)** y<br>y= result of operation 2 i.e 121 121<br><br>**Operation 4 (Post increment)**<br>n=n+1 ( post increment having least priority)<br>Memory n looks like<br>n<br>~~10~~ ~~11~~ 12<br><br>**Final Value of memories:**<br>n➔ 12 and y➔121 |

**Practice Consider 2 programs and try to find out output**

```
#include "iostream"
using namespace std;
void main()
{
	int n = 10;

int y = n++ + ++n * ++n - n-- - --n;
	cout << n << endl;
	cout << y << endl;
}
```

Output

```
#include "iostream"
using namespace std;
void main()
{
	int n = 10;
	cout << "Value of n = "
		<< n << endl;

	cout << "Post Increment on n = "
		<< n++ << endl;
	cout << "After Post Increment "
		<< n << endl;
	cout << "Pre increment on n "
		<< ++n << endl;
	cout << "After Pre Increment "
		<< n << endl;
	cout << "Post Decrement "
		<< n-- << endl;
	cout << "After Post Decrement "
		<< n << endl;
	cout << "Pre Decrement "
		<< --n << endl;
	cout << "After Pre Decrement "
		<< n << endl;

}
```

Output

# Northern University, Nowshera

**Programming Fundamentals (CS-)**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776

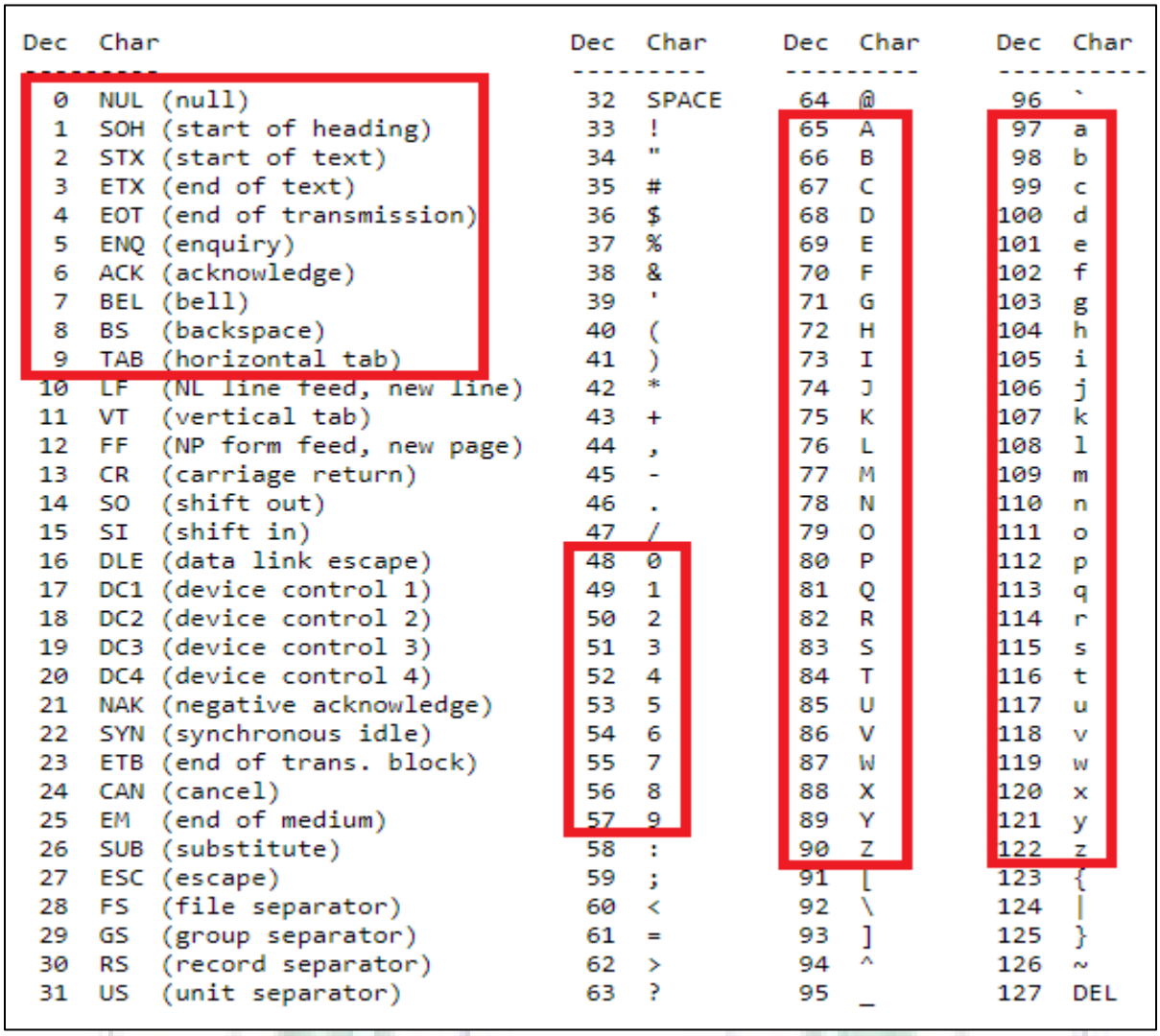
| | |
|---|---|
| | |

Now let us talk about char data type. Before moving onto a program we will first be discussing ASCII table. Every char is assigned some unique integer number called ASCII code. Following is the list of characters and their ASCII values. ASCII (American Standard Code for Information Interchange)

# Northern University, Nowshera

**Programming Fundamentals (CS-)**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776



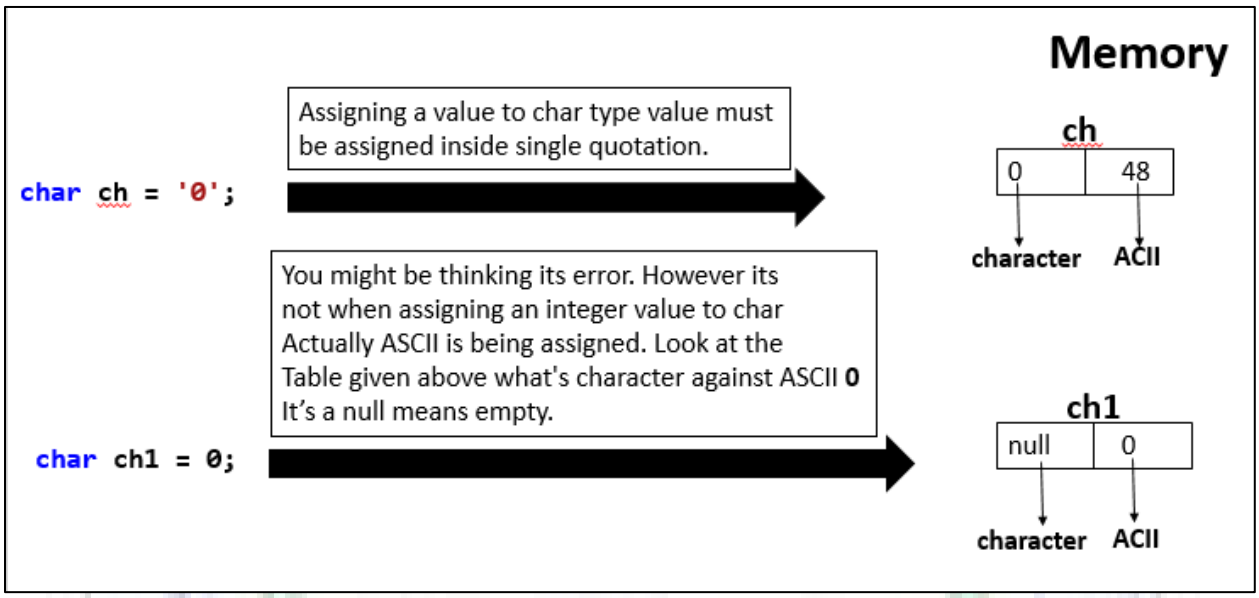
| Dec | Char | Dec | Char | Dec | Char | Dec | Char |
|---|---|---|---|---|---|---|---|
| 0 | NUL (null) | 32 | SPACE | 64 | @ | 96 | ` |
| 1 | SOH (start of heading) | 33 | ! | 65 | A | 97 | a |
| 2 | STX (start of text) | 34 | " | 66 | B | 98 | b |
| 3 | ETX (end of text) | 35 | # | 67 | C | 99 | c |
| 4 | EOT (end of transmission) | 36 | $ | 68 | D | 100 | d |
| 5 | ENQ (enquiry) | 37 | % | 69 | E | 101 | e |
| 6 | ACK (acknowledge) | 38 | & | 70 | F | 102 | f |
| 7 | BEL (bell) | 39 | ' | 71 | G | 103 | g |
| 8 | BS (backspace) | 40 | ( | 72 | H | 104 | h |
| 9 | TAB (horizontal tab) | 41 | ) | 73 | I | 105 | i |
| 10 | LF (NL line feed, new line) | 42 | * | 74 | J | 106 | j |
| 11 | VT (vertical tab) | 43 | + | 75 | K | 107 | k |
| 12 | FF (NP form feed, new page) | 44 | , | 76 | L | 108 | l |
| 13 | CR (carriage return) | 45 | - | 77 | M | 109 | m |
| 14 | SO (shift out) | 46 | . | 78 | N | 110 | n |
| 15 | SI (shift in) | 47 | / | 79 | O | 111 | o |
| 16 | DLE (data link escape) | 48 | 0 | 80 | P | 112 | p |
| 17 | DC1 (device control 1) | 49 | 1 | 81 | Q | 113 | q |
| 18 | DC2 (device control 2) | 50 | 2 | 82 | R | 114 | r |
| 19 | DC3 (device control 3) | 51 | 3 | 83 | S | 115 | s |
| 20 | DC4 (device control 4) | 52 | 4 | 84 | T | 116 | t |
| 21 | NAK (negative acknowledge) | 53 | 5 | 85 | U | 117 | u |
| 22 | SYN (synchronous idle) | 54 | 6 | 86 | V | 118 | v |
| 23 | ETB (end of trans. block) | 55 | 7 | 87 | W | 119 | w |
| 24 | CAN (cancel) | 56 | 8 | 88 | X | 120 | x |
| 25 | EM (end of medium) | 57 | 9 | 89 | Y | 121 | y |
| 26 | SUB (substitute) | 58 | : | 90 | Z | 122 | z |
| 27 | ESC (escape) | 59 | ; | 91 | [ | 123 | { |
| 28 | FS (file separator) | 60 | < | 92 | \ | 124 | \| |
| 29 | GS (group separator) | 61 | = | 93 | ] | 125 | } |
| 30 | RS (record separator) | 62 | > | 94 | ^ | 126 | ~ |
| 31 | US (unit separator) | 63 | ? | 95 | _ | 127 | DEL |

In the figure given above some character and their codes are highlighted. Consider following statements

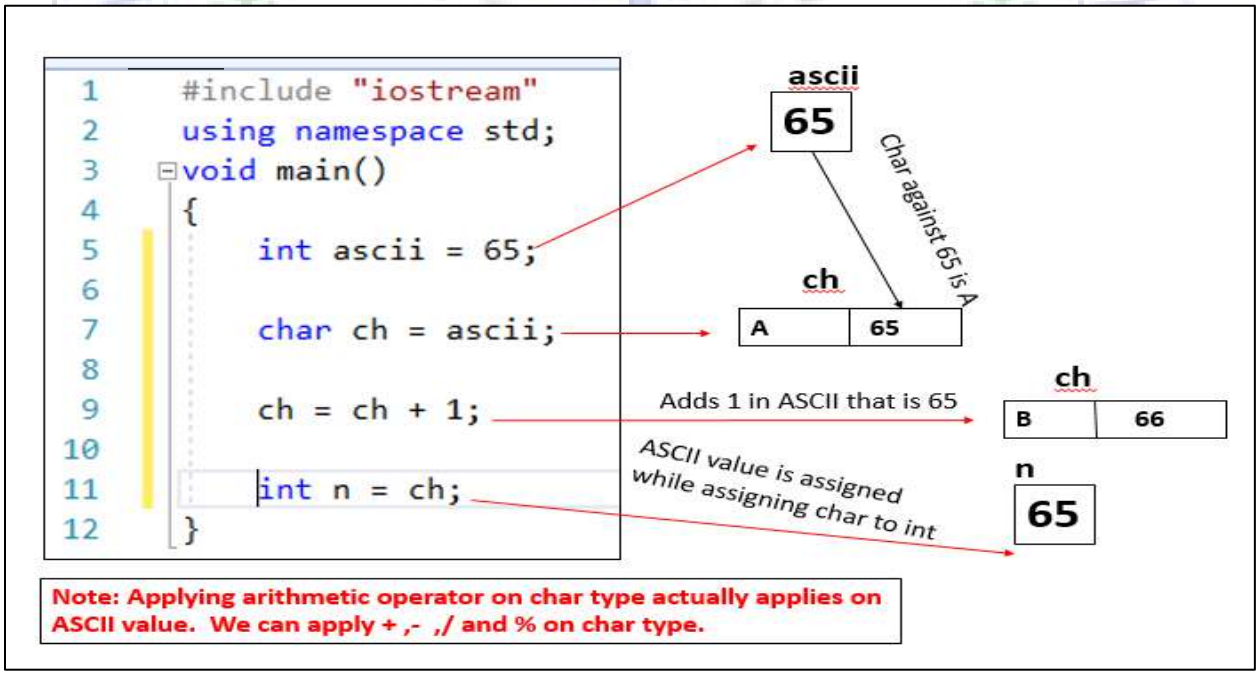
# Northern University, Nowshera

**Programming Fundamentals (CS-)**

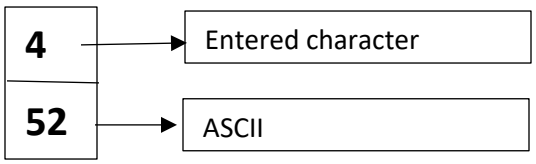
Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776




Consider following program for understanding char in more detail.




char ch;cin>>ch; // assume that user enters 4 than this 4 is consider as character and in above table you can see ASCII value of 4. Memory will look as.



# Northern University, Nowshera

**Programming Fundamentals (CS-)**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776

## Lecture # 8 Integer and Char

Objective of this lecture is to make you more familiar with integer and char type. For detailed understanding let us look at the scenario given below and try to code for it.

**Program:** A bowler bowls an over of 6 balls without taking any wicket. You have to input number of runs scored on each bowl. Calculate and display total runs scored. ***Assuming that values entered will either be 0,1,2,3,4 and 6.***

We shall be dealing with four different version of programs for above statement.

### Version 1

| Steps | Code |
|---|---|
| 1. Declare 6 integer type memories for storing runs scored.<br>2. Input in above memories.<br>3. Declare another memory for storing total runs<br>4. Add values of all 6 memories declared in part 1 and store in memory declared in part 3.<br>5. Display Memory declared in part 3. | `#include "iostream"`<br>`using namespace std;`<br>`void main()`<br>`{`<br>`int b1_score, b2_score, b3_sore, b4_score, b5_score, b6_score;`<br><br>`cout << "Enter score on ball 1,2,3,4,5 and 6 ";`<br>`cin >> b1_score >> b2_score >> b3_sore >> b4_score >> b5_score >> b6_score;`<br><br>`int total;`<br>`total = b1_score + b2_score + b3_sore + b4_score + b5_score + b6_score;`<br><br>`cout << "Total Score " << total << endl;`<br>`}` |

How many bytes of memory will be consumed? We declared 7 integer type memories. If 1 memory contains 4byte than total memory consumed is i.e 7*4= 28byts. It size can be reduced to half by using short type. Can above program be converted into some other way so that memory size get reduced further?

If we think of maximum runs scored on ball is 6.

So why to use int if a char is capable of storing 6.

And Maximum runs can be scored in over is 36(which cannot be stored in char so for total runs we should use int/short (short preferred)

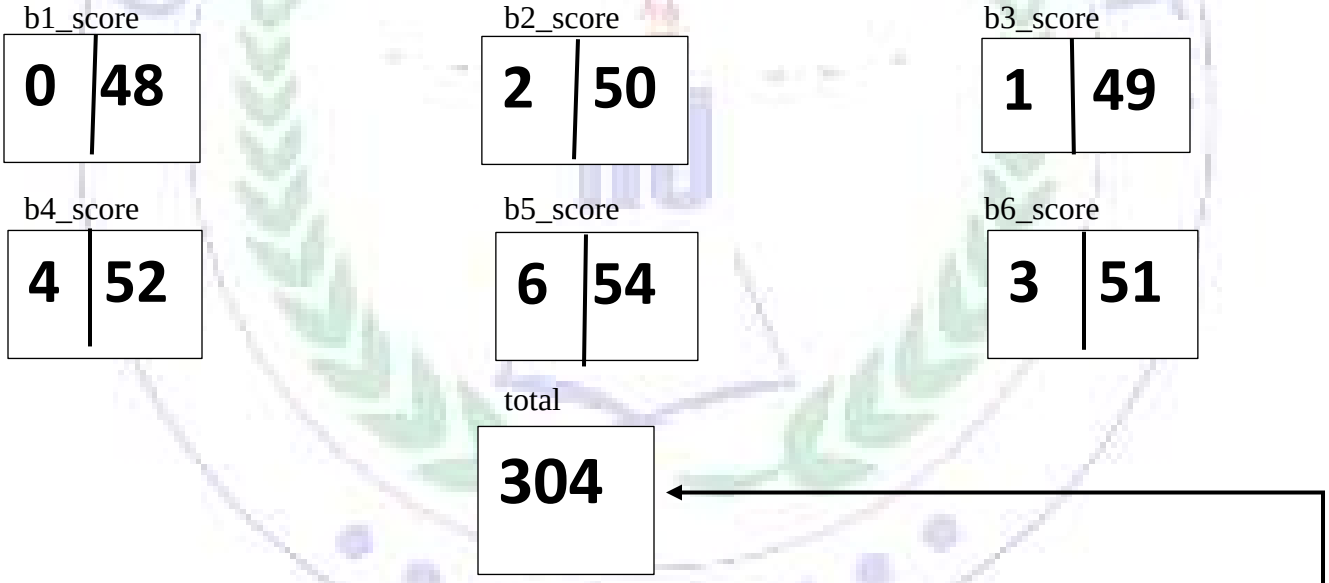
# Northern University, Nowshera

**Programming Fundamentals (CS-)**

M **Version 2** ır Email:athar@northern.edu.pk WahatsApp: 0333-50776

```
1   #include "iostream"
2   using namespace std;
3   void main()
4   {
5       char  b1_score, b2_score, b3_sore, b4_score, b5_score, b6_score;
6
7       cout << "Enter score on ball 1,2,3,4,5 and 6 ";
8       cin >> b1_score;
9       cin >> b2_score;
10      cin >> b3_sore;
11      cin >> b4_score;
12      cin >> b5_score;
13      cin>> b6_score;
14
15      int total;
16      total = b1_score + b2_score + b3_sore + b4_score + b5_score + b6_score;
17
18      cout << "Total Score " << total << endl;
19  }
```

Memory



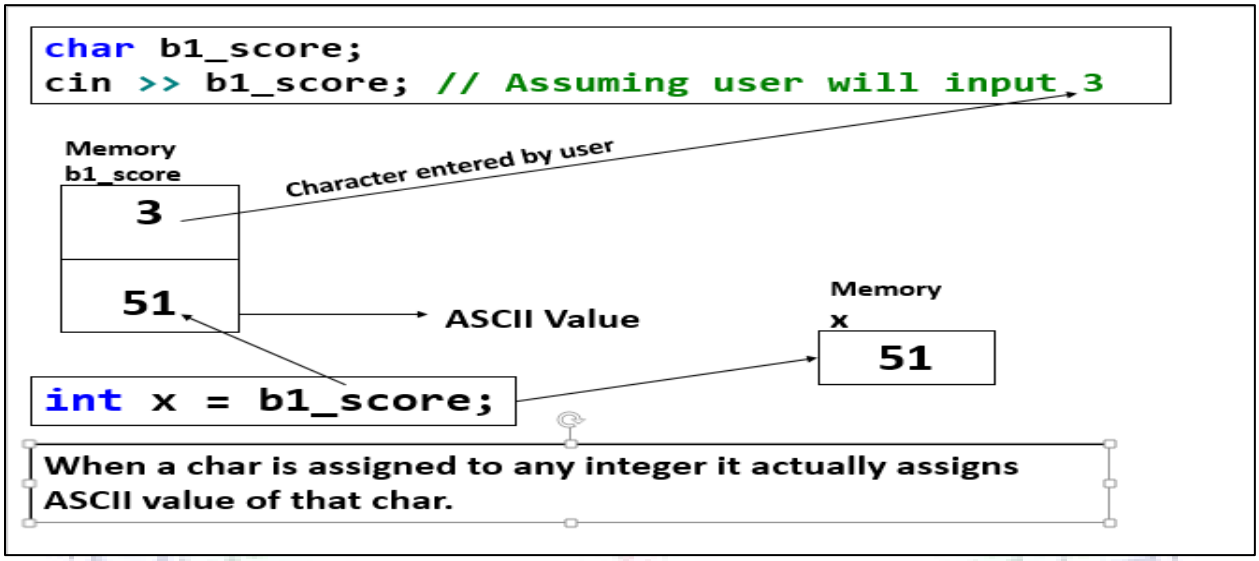

As we discussed in previous lecture that whenever an arithmetic operation is performed it is actually applied on ASCII. In above char memories single digits are char and along with them are ASCII values

```
total = b1_score + b2_score + b3_sore + b4_score + b5_score + b6_score;
```

**48 + 50 + 49 + 52 + 54 + 51 ➔ 304**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776


***Answer is wrong because maximum runs in an over can be 36 this is because in case of char + operator adds asci values. Now consider logic given below and will change above statement accordingly.***




What if we want to store actual character in above case? Now consider table given below.

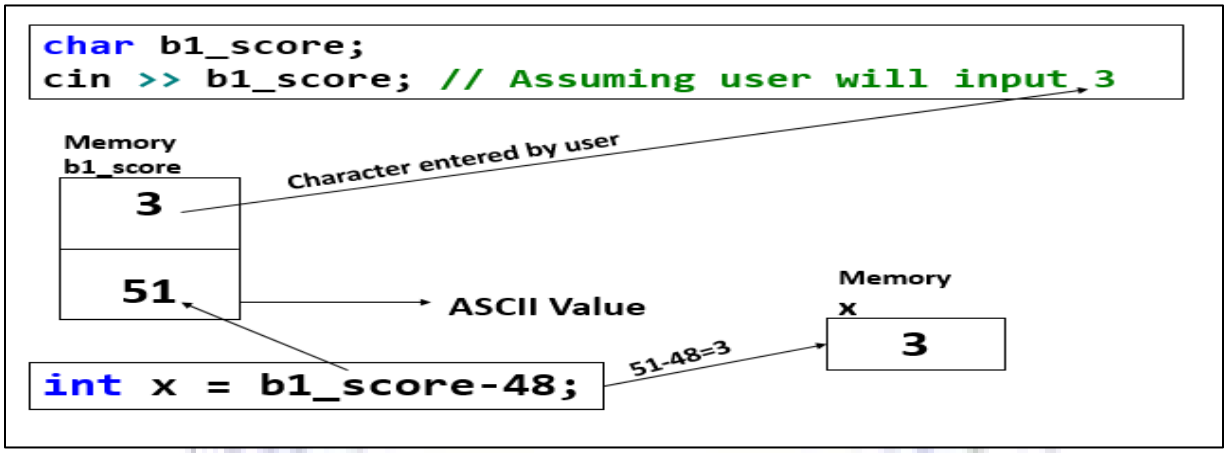
| Char | ASCII | Subtract 48 from ASCII |
|---|---|---|
| 0 | 48 | 48 - 48 = 0 |
| 1 | 49 | 49 - 48=1 |
| 2 | 50 | 50 - 48=2 |
| 3 | 51 | 51 - 48=3 |
| 4 | 52 | 52 - 48=4 |
| 5 | 53 | 53 - 48=5 |
| 6 | 54 | 54 - 48=6 |
| 7 | 55 | 55 - 48=7 |
| 8 | 56 | 56 - 48=8 |
| 9 | 57 | 57 - 48=9 |

From above table it is obvious if char entered by use is between 0-9 we can subtract 48 to get original char.

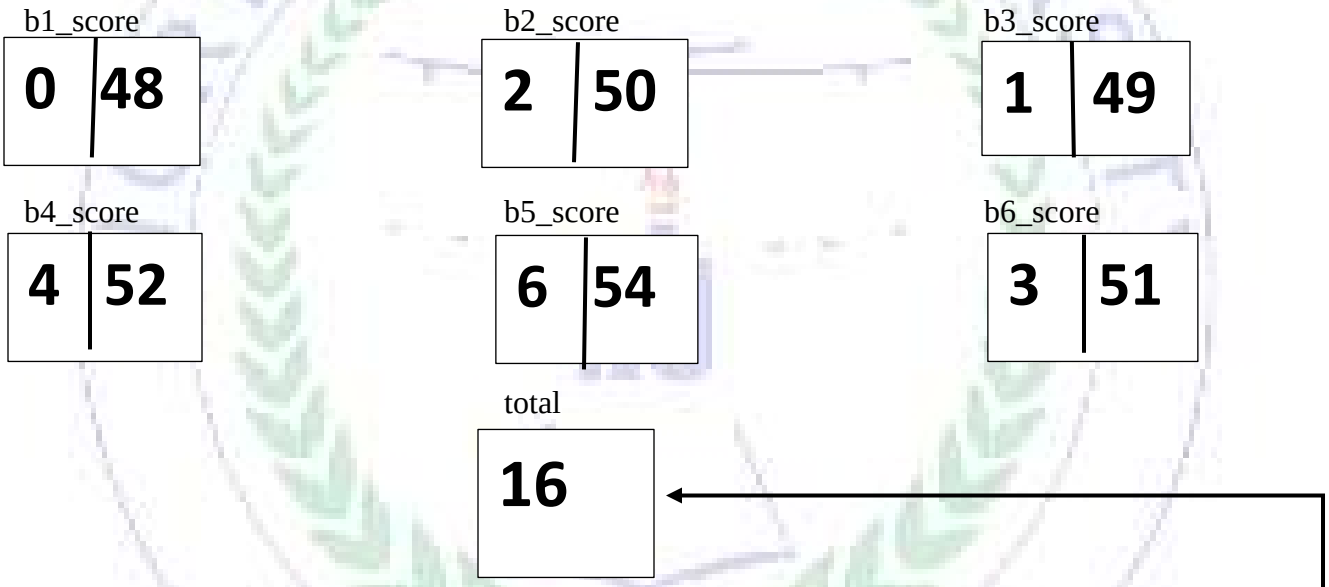
# Northern University, Nowshera

**Programming Fundamentals (CS-)**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776




Memory




As we discussed in previous lecture that whenever an arithmetic operation is performed it is actually applied on ASCII. In above char memories single digits are char and along with them are ASCII values

```
total = b1_score + b2_score + b3_sore + b4_score + b5_score + b6_score;
```

**(48- 48)+ (50-48) + (49-48)+ (52-48) + (54-48)+(51-48)➔ 16**

Now total variable will have correct value.

**Version 3:**

Why to declare so many variables we can do it by declaring only two integer variables.

# Northern University, Nowshera

**Programming Fundamentals (CS-)**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776

```
1   #include "iostream"
2   using namespace std;
3   void main()
4   {
5       int total = 0, score;
6       cout << "Enter runs scored on ball 1,2,3,4,5 and 6 ";
7       cin >> score;
8       total = total + score;
9
10      cin >> score;
11      total = total + score;
12
13      cin >> score;
14      total = total + score;
15
16      cin >> score;
17      total = total + score;
18
19      cin >> score;
20      total = total + score;
21
22      cin >> score;
23      total = total + score;
24
25      cout << total;
26  }
```

Memory values

| Line # | Value of total | Value of score |
|---|---|---|
| 5 | total is assigned 0 | Has garbage value |
| 7 | 0 | User enter 2 |
| 8 | 2 (0+2) | 2 |
| 10 | 2 | User enters 3 |
| 11 | 5 (2+3) | 3 |
| 13 | 5 | User enter 0 |
| 14 | 5 (5+0) | 0 |
| 16 | 5 | User enter 6 |
| 17 | 11 (5+6) | 6 |
| 19 | 11 | User enter 2 |
| 20 | 13 (11+2) | 2 |
| 21 | 13 | User enter 4 |
| 22 | 17 (13+4) | 4 |

Final value of total i.e 17 will be displayed on screen.

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776


Same program can be done using char type also.

**Version 4:**

```
#include "iostream"
using namespace std;
void main()
{
    int total = 0;
    char score;
    cout << "Enter runs scored on ball 1,2,3,4,5 and 6 ";
    cin >> score;
    total = total + (score-48);  // subtracting 48 to get original character

    cin >> score;
    total = total + (score-48);

    cin >> score;
    total = total + (score-48);

    cin >> score;
    total = total + (score-48);

    cin >> score;
    total = total + (score-48);

    cin >> score;
    total = total + (score-48);

    cout << total;
}
```

# Northern University, Nowshera

**Programming Fundamentals (CS-)**

Muhammad Athar Email:athar@northern.edu.pk WahatsApp: 0333-50776