11. Characters and ASCII codes

- Shivam Malhotra

Characters

- Characters are internally represented by integral codes, called ASCII code. For example, ASCII code for 'A' is 65.

- In the last lecture, we discussed how we can print the ASCII code for a input character

```
cin >> ch;              // Take input from user
int code = ch;          // Store in integer variable
cout << code;           // Print the ASCII code
```

Characters

- Using the same program, you can check that the following are the ASCII values for the common characters :

Characters	ASCII Values
'A' - 'Z'	65-90
'a' - 'z'	97-122
'0' - '9'	48-57

Characters

- The ranges of character codes are continuous. For example,

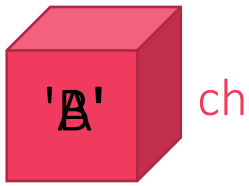
  If ASCII code of 'A' = 65 ➡ ASCII code of 'B' = 66

  If ASCII code of '1' = 49 ➡ ASCII code of '3' = 51

- Since characters are internally represented by integral codes, hence we can perform mathematical operations like + on characters.

```
char ch = 'A';
ch = ch + 1;
cout << ch;
```

Output :
B



Practice

Write a program which takes a lower case character as input and prints the corresponding upper case character.

Input :

a

Output :

A

Input :

d

Output :

D

Solution Scheme

- Let the input be character 'd', whose ASCII code is 100

- Consider the following code

```
int  val1 = 'd';              // val1 = 100
int  val2 = 'a';              // val2 = 97
int  val3 = val1 - val2;      // val3 = 3
char ch  = 'A' + val3;        // ch = 'D'
```

Practice

Write a program which takes a lower case character as input and prints the corresponding upper case character.

```
#include <iostream.h>
#include <conio.h>
int main()
{
        clrscr();
        char inputCh;
        cout << "Enter character : ";
        cin >> inputCh;
        int val1 = inputCh;
        int val2 = 'a';
        int val3 = val1 - val2;
        char outputCh = 'A' + val3;
        cout << "Output character : " << outputCh;
        getch();
        return 0;
}
```

Character Data Type

- You can simply use it as **char.** For example,

```
char someName;
```

Data Type	Approximate size (in bytes)	Range
char	1	-128 to 127
unsigned char	1	0 to 255
signed char	1	Same as char

What’s ahead?

In the next video, we will study about constants and arithmetic operators