Programming Fundamentals

Dr. Abdul Aziz

Assistant Professor & HoD

Department of Software Engineering

NUCES-FAST, Karachi.

Week 3, Class 1

Escape Sequence

- The purpose of the escape sequence is to represent the characters that cannot be used normally using the keyboard

\a	Alarm or Beep	It is used to generate a bell sound in the C program.
\b	Backspace	It is used to move the cursor one place backward.
\f	Form Feed	It is used to move the cursor to the start of the next logical page.
\n	New Line	It moves the cursor to the start of the next line.
\r	Carriage Return	It moves the cursor to the start of the current line.
\t	Horizontal Tab	It inserts some whitespace to the left of the cursor and moves the cursor accordingly.
\v	Vertical Tab	It is used to insert vertical space.
\\	Backlash	Use to insert backslash character.
\'	Single Quote	It is used to display a single quotation mark.
\"	Double Quote	It is used to display double quotation marks.
\?	Question Mark	It is used to display a question mark.
\ooo	Octal Number	It is used to represent an octal number.
\xhh	Hexadecimal Number	It represents the hexadecimal number.
\0	NULL	It represents the NULL character.

Program 1

- Write a program to take an integer value as an input from user and print its table from 1 to 5.
- Required : IPO and code.
- Output:
  - 3 x 1 = 3
  - 3 x 2 = 6
  - 3 x 3 = 9

IPO program 1

Data	Process	Output
int table; to store the user input	1: print prompt to input the table user want to print. 2: Take input and save in table variable. 3: print output 5 times by multiplying table with 1 to 5. 4: end.	3 x 1 = 3 3 x 2 = 6 3 x 3 = 9 3 x 3 = 12 3 x 3 = 15

Program 2

- Write a program that takes 3 subject marks as input and print its total, marks obtain and percentage.
- Required : IPO and code
- Output:
  - Subject 1 = 70 /100
  - Subject 2 = 60 / 100
  - Subject 3 = 50 /100
  - Total Marks obtain = 180 / 300
  - Total percentage = 60 %

IPO program 2

Data	Process	Output
• 3 float variables to store subject marks. s1, s2 and s3. • 1 float for total. • 1 float for percentage.	1: prompt user to input 3 subject marks. 2: Take input and save in s1, s2 and s3 variable. 3: calculate total marks and save in total variable by adding s1, s2 and s3. 4: calculate percentage and save in variable. Total*100/300. 5: print total 6: print percentage 7: end.	Subject 1 = 70 /100 Subject 2 = 60 / 100 Subject 3 = 50 /100 Total Marks obtain = 180/300 Total percentage = 60 %

Week 3, Class 2

Operators in C

- Arithmetic Operator
- Arithmetic Assignment
- Increment / Decrement
- Relational
- Logical
- Conditional
- Bitwise
- Special

Arithmetic Operator

- +
- -
- *
- /
- %

Assignment Operators

- =
- *=
- /=
- +=
- -=
- %=

Increment / Decrement Operator

- ++
- --

Relational Operator

Operator	Meaning
<	is less than
<=	is less than or equal to
>	is greater than
>=	is greaten than or equal to
= =	is equal to
! =	is not equal to

Logical Operator

& &	meaning	logical AND
||	meaning	logical OR
!	meaning	logical NOT

Week 3, Class 3

Condition Operator

- ? (Ternary Operator)

Bitwise Operator

Operator	Meaning
&	bitwise AND
|	bitwise OR
^	Bitwise exclusive OR
<<	shift left
>>	shift right
~	One's complement

```
#include <stdio.h>
main() {
int p = 60; /* 60 = 0011 1100 */
int q = 13; /* 13 = 0000 1101 */
int r = 0;
r = p | q; /* 61 = 0011 1101 */
        printf("Line 1 – The value of r is %d\n", r );
r = p & q; /* 12 = 0000 1100 */
        printf("Line 2 – The value of r is %d\n", r );
r = ~p; /*-61 = 1100 0011 */
        printf("Line 3 – The value of r is %d\n", r );
r = p ^ q; /* 49 = 0011 0001 */
        printf("Line 4 – The value of r is %d\n", r );
r = p >> 2; /* 15 = 0000 1111 */
        printf("Line 5 – The value of r is %d\n", r );
r = p << 2; /* 240 = 1111 0000 */
        printf("Line 6 – The value of r is %d\n", r );
}
```

Example

Special Operator

- , Comma Operator
- & Address Operator
- Sizeof Operator
- * and & Pointer Operator
- . and -> Member Selection Operator
- #

math.h

- The **math.h** header defines various C mathematical functions
- **Example:**
- **double ceil (double x)**
- **double floor(double x)**
- **double fabs(double x)**
- **double log(double x)**
- **double log10(double x)**
- **double sqrt(double x)**

Example

```
#include <math.h>
#include <stdio.h>
void main( void ){
        float val1 = 1.6, val2= 1.2;
        printf("value1 = %.1lf\n", ceil(val1));
        printf("value2 = %.1lf\n", ceil(val2));
}
```

Example

```
void main()
{
    double x, ret;
    x = 2.7;

    /* finding log(2.7) */
    ret = log(x);

   printf("log(%lf) = %lf", x, ret);

    return (0);
}
```

OUTPUT:

```
log(2.700000) = 0.993252
```