# Escape Sequence:

The backslash (\) is called an escape character. When encountering a backslash in a string, the compiler looks ahead at the next character and combine it with the backslash to form an escape sequence

|  |  |
| --- | --- |
| Escape Sequence | Description |
| \n | Newline.  Position the cursor at the beginning of the next line |
| \t | Horizontal Tab  Position the cursor at the beginning of the next line |
| \a | Alert  Produces a sound or visible alert without changing the current cursor position |
| \\ | Backslash  Inserts a backslash character in a string |
| \” | Double quote  Inserts a double-quote character in a string |

## Example:

### Program 1:

int main(int argc, char \*argv[]) {

printf("Hello World!\\\\");

printf("Programming is fun!");

return 0;

}

This program will print

Hello world!\\Programming is fun!

### Program 2:

int main(int argc, char \*argv[]) {

printf("Hello World!\"");

printf("Programming is fun!\"");

return 0;

}

This program will print

Hello World!”programming is fun!”

### Program 3:

int main(int argc, char \*argv[]) {

printf("programming\nis\nfun\n!");

return 0;

}

This program will print

Programming

is

fun

!

# Output Formatting using printf

|  |  |
| --- | --- |
| Format Specifier | Argument printed as |
| %d, %i | Integer |
| %o | Octal Integer |
| %x, %X | Hexadecimal integer |
| %u | Unsigned integer |
| %c | Character |
| %s | String |
| %f | Floating point in the format m.dddddd |
| %e, %E | Scientific notation format eg. 3.5e10 |
| %g, %G | %f or %e/%E, which ever is shorter |
| %p | Pointer (implementation dependent) |
| %% | The % character |

## Examples

### Program 1:

int main(int argc, char \*argv[]) {

printf("The number is: %d", 9);

return 0;

}

The program prints:

The number is: 9

### Program 3:

int main(int argc, char \*argv[]) {

printf("%s C is amazing", "Programming in");

return 0;

}

The program will print:

Programming in C is amazing

# Precedence and Order of Evaluation:

|  |  |
| --- | --- |
| Operator | Associativity |
| () | left to right |
| ++ -- | right to left |
| \* / % | left to right |
| + - | left to right |
| < <= > >= | left to right |
| == != | left to right |
| & | left to right |
| ^ | left to right |
| && | left to right |
| || | left to right |
| = += -= \*= /= %= &= |= <<= >>= | right to left |