Standard Input and Output in C Programming

Input and Output (I/O) operations are fundamental components of any programming language. In C, these operations are handled through the standard input/output library functions.

Standard I/O Functions

C provides several standard I/O functions within the library **stdio.h** to interact with the console and files. These include printf(), which is used for formatted output, scanf(), which is used for formatted input, getchar(), for reading a single character, putchar(), for writing a single character, puts(), for writing a string with a newline, and gets(), for reading a line of text.

printf() Function

The **printf()** function is used to print formatted data to the standard output, typically the console. It takes a format string and a list of variables, where the format string contains placeholders for the variables. For example, **printf("Hello, %s!\n", "World");** will print "Hello, World!".

Format Specifiers

Format specifiers like **%d, %f, %c, %s, and %x** are used in the format string of printf() to specify the type of data to be printed. For instance, %d is used for integers, %f for floats, %c for characters, and so on.

Escape Sequences

Escape sequences like **\n, \t, \\, and** \" allow for special characters in strings. For example, **printf("Hello\nWorld");** will print "Hello" on one line and "World" on the next.

scanf() Function

The **scanf()** function reads formatted input from the standard input (usually the keyboard). It takes a format string and a list of variables, similar to printf(). For example, **scanf("%d", &num);** reads an integer into the variable num.

Reading and Writing Characters

getchar() reads a single character from standard input, while **putchar()** writes a single character to standard output. These functions are useful when dealing with individual characters, such as in menu selections or simple prompts.

Writing Strings and Reading Lines

puts() writes a string to standard output, automatically adding a newline character. It's convenient for displaying complete strings. **gets()** is used for reading entire lines of text, which is common in text-based user interfaces.

Example :

Write a C program that allows the user to enter the value of a circle's radius, calculates its area, and displays it on the screen.

```c
#include <stdio.h>
#include <math.h>

int main() {
    // Declare variables
    float radius, area;

    // Prompt user for input
    printf("Enter the radius of the circle: ");
    scanf("%f", &radius);

    // Calculate area using pow() function
    area = M_PI * pow(radius, 2);

    // Display the area
    printf("The area of the circle with radius %.2f is %.2f\n", radius, area);

    return 0;
}
```