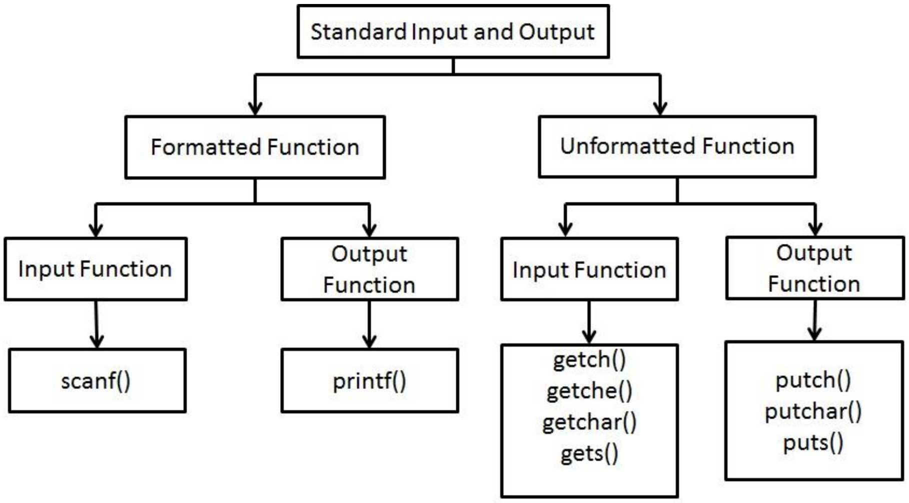
Input Output Operation in C

- In C programming,
  Input operations are used to read data from input devices, such as the keyboard.
  Output operations are used to write data to output devices, such as the screen or a file.
- There are different types of I/O operations in C:
  1. Formatted I/O
  2. Unformatted I/O



1. Formatted I/O

- The formatted functions basically present or accept the available data (input) in a specific format. In C programming, printf() and scanf() functions are used.

printf() and scanf()

- The printf() function is used to output data to the console.
- The scanf() functions is used to read the input from the console.
- Example:

```
#include <stdio.h>
int main()
{

    int num1, num2, sum;

    printf("Enter two integers: ");
    scanf("%d %d", &num1, &num2);

    sum = num1 + num2;

    printf("%d + %d = %d", num1, num2, sum);
    return 0;
}
```

2. Unformatted I/O

- These functions are called unformatted I/O functions because we cannot use format specifiers in these functions.
- Unformatted I/O functions only work with character data types or arrays/strings and not with other data types.
- The unformatted I/O functions in C are:

a. getch() and putch()

b. getchar() and putchar()

c. getch() and putch()

d. getc() and putc()

a. getch() and putch()

- The **getch()** function is used to read a single character from the console.
- Input data is not displayed to the console.
- The **putch()** function is used to output a single character to the console.
- Example:

```
#include <stdio.h>
#include <conio.h>

int main()
{
    char ch;

    printf("Enter a character: ");
    ch = getch();
    printf("\nYou entered: ");
    putch(ch);

    return 0;
}
```

b. getchar() and putchar()

- The **getchar()** function is used to read a single character from the console
- The input data is displayed on the screen.
- The **putchar()** function is used to output a single character to the console.
- Example:

```
#include <stdio.h>

int main()
{
    char ch;

    printf("Enter a character: ");
    ch = getchar();

    printf("\nYou entered: ");
    putchar(ch);

    return 0;
}
```

c. gets() and puts()

- The **gets()** function is used to read a string.
- The **puts()** function is used to output a string.
- Example:

```
#include <stdio.h>

int main()
{
    char str[100];

    printf("Enter a string: ");
    gets(str);
    printf("You entered: ");
    puts(str);

    return 0;
}
```

d. getc() and putc()

- The getc(), putc() functions are file handling function in C programming language.
- **getc()** is used to read a character from a file.
- **putc()** is used to write into a file.
- Example:

```
#include <stdio.h>

int main()
{
    FILE *fp;
    char ch;

    fp = fopen("test.txt", "r");

    while ((ch = getc(fp)) != EOF)
    {
        putc(ch, stdout);
    }

    fclose(fp);
    return 0;
}
```