**getc():**  
It reads a single character from a given input stream and returns the corresponding integer value (typically ASCII value of read character) on success. It returns EOF on failure.

Syntax:

int getc(FILE \*stream);

**EXAMPLE**

|  |
| --- |
| // Example for getc() in C  #include <stdio.h>  int main()  {     printf("%c", getc(stdin));     return(0);  } |

Input: g (press enter key)

Output: g   
  
  
**getchar():**  
The difference between getc() and getchar() is getc() can read from any input stream, but getchar() reads from standard input. So getchar() is equivalent to getc(stdin).

Syntax:

int getchar(void);

**EXAMPLE**

|  |
| --- |
| // Example for getchar() in C  #include <stdio.h>  int main()  {     printf("%c", getchar());     return 0;  }  **EXAMPLE** |

Input: g(press enter key)

Output: g

|  |  |
| --- | --- |
| getc() | Declaration: int getc(FILE \*fp)  getc functions is used to read a character from a file. In a C program, we read a character as below. **getc** (fp); |
| putc() | Declaration: int putc(int char, FILE \*fp)  putc function is used to display a character on standard output or is used to write into a file. In a C program, we can use putc as below.  putc(char,stdout); putc(char, fp); |

#include <stdio.h>

int main()

{

char ch;

FILE \*fp;

if (fp = fopen("test.c", "r"))

{

ch = getc(fp);

while (ch != EOF)

{

putc(ch, stdout);

ch = getc(fp);

}

fclose(fp);

return 0;

}

return 1;

}

**Output:**

|  |
| --- |
| Hi, How are you? |

The **putchar(int char)** method in C is used to write a character, of unsigned char type, to stdout. This character is passed as the parameter to this method.

**Syntax:**

int putchar(int char)

|  |
| --- |
| // C program to demonstrate putchar() method    #include <stdio.h>    int main()  {        // Get the character to be written      char ch = 'G';        // Write the Character to stdout      putchar(ch);        return (0);  } |

**Output:**

G