A **function pointer** is a pointer variable that can store address of a function and then using the function pointer we can call initialized function in our program.

Function Pointer Example using C program

Declaration of **function pointer**

|  |  |
| --- | --- |
| 1 | return\_type (\*fun\_pointer\_name)(argument\_type\_list); |

Initialization of **function pointer**

|  |  |
| --- | --- |
| 1 | fun\_pointer\_name= &function\_name; |
| 1 | fun\_pointer\_name= function\_name; |

Calling of **function pointer**

|  |  |
| --- | --- |
| 1 | \*(fun\_pointer\_name)(actual\_argument\_list); |

Declaration with Initialization

|  |  |
| --- | --- |
| 1 | return\_type (\*fun\_pointer\_name)(argument\_type\_list)= &function\_name; |

**Consider the example**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | /\*function pointer example in c.\*/    #include <stdio.h>      //function: sum, will return sum of two  //integer numbers  int addTwoNumbers(int x,int y)  {      return x+y;  }    int main()  {      int a,b,sum;        //function pointer declaration      int (\*ptr\_sum)(int,int);      //function initialisation      ptr\_sum=&addTwoNumbers;        a=10;      b=20;        //function calling      sum=(\*ptr\_sum)(a,b);        printf("Sum is: %d\n",sum);        return 0;  } |

Sum is: 30

**int addTwoNumbers(int x,int y);**

This function will take two integer arguments and returns addition of those numbers.

**int (\*ptr\_sum)(int,int);**

This is the declaration of the function pointer for **addTwoNumbers** function.

**ptr\_sum=&addTwoNumbers;**

This statement is initializing the function pointer with address of function **addTwoNumbers**.

**(\*ptr\_sum)(a,b);**

This statement is the calling the function **ptr\_sum** and **ptr\_sum** will point to the **addTwoNumbers**.