SNS COLLEGE OF TECHNOLOGY

Coimbatore-35.
An Autonomous Institution

COURSE NAME : 19CST101 PROGRAMMING FOR PROBLEM SOLVING

I YEAR/ I SEMESTER

UNIT-II C PROGRAMMING BASICS

Topic: Input and Output Operations

Ms. Sumathi B
Assistant Professor
Department of Computer Science and Engineering

C Input and Output (I/O)

As we all know the three essential functions of a computer are reading, processing and writing data. Majority of the programs take data as input, and then after processing the processed data is being displayed which is called information. In C programming you can use `scanf()` and `printf()` predefined function to read and print data.

Managing Input/Output

I/O operations are useful for a program to interact with users. `stdlib` is the standard C library for input-output operations. While dealing with input-output operations in C, two important streams play their role. These are:

1. Standard Input (stdin)
2. Standard Output (stdout)

`Standard input` or `stdin` is used for taking input from devices such as the keyboard as a data stream. `Standard output` or `stdout` is used for giving output to a device such as a monitor. For using I/O functionality, programmers must include `stdio header-file` within the program.

C Output Functions

C programming language provides built-in functions to perform output operation. The output operations are used to display data on user screen (output screen) or printer or any file. The c programming language provides the following built-in output functions...

1. printf()
2. putchar()
3. puts()
4. fprintf()

printf() function

The printf() function is used to print string or data values or a combination of string and data values on the output screen (User screen). The printf() function is built-in function defined in a header file called "**stdio.h**". When we want to use printf() function in our program we need to include the respective header file (stdio.h) using the **#include** statement. The printf() function has the following syntax...

C Output Functions

Syntax:

```
printf("message to be display!!!");
```

The printf() function is also used to display data values. When we want to display data values we use **format string** of the data value to be displayed.

Syntax:

```
printf("format string",variableName);
```

The printf() function can also be used to display string along with data values.

Syntax:

```
printf("String format string",variableName);
```

C Output Functions

Example 1: C Output

```c
#include <stdio.h>
int main()
{
    // Displays the string inside quotations
    printf("C Programming");
    return 0;
}
```

Output

```
C Programming
```

C Output Functions

Example 2: Integer Output

```c
#include <stdio.h>
int main()
{
    int testInteger = 5;
    printf("Number = %d", testInteger);
    return 0;
}
```

Output

```
Number = 5
```

We use `%d` format specifier to print `int` types. Here, the `%d` inside the quotations will be replaced by the value of `testInteger`.

C Output Functions

Example 3: float and double Output

```c
#include <stdio.h>
int main()
{
    float number1 = 13.5;
    double number2 = 12.4;

    printf("number1 = %f\n", number1);
    printf("number2 = %lf", number2);
    return 0;
}
```

Output

```
number1 = 13.500000
number2 = 12.400000
```

To print `float`, we use `%f` format specifier. Similarly, we use `%lf` to print `double` values.

C Output Functions

Example 4: Print Characters

```c
#include <stdio.h>
int main()
{
    char chr = 'a';
    printf("character = %c", chr);
    return 0;
}
```

Output

```
character = a
```

To print `char`, we use `%c` format specifier.

C Output Functions

Formatted printf() function

Generally, when we write multiple printf() statements the result is displayed in a single line because the printf() function displays the output in a single line. Consider the following example program...

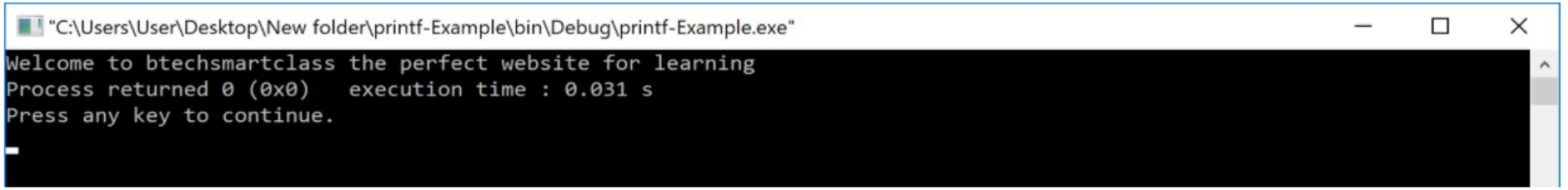
Example Program

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

void main(){
   printf("Welcome to ");
   printf("btechsmartclass ");
   printf("the perfect website for learning");

}
```

Output:



In the above program, there are 3 printf() statements written in different lines but the output is displayed in single line only.

C Output Functions

To display the output in different lines or as we wish, we use some special characters called **escape sequences**. Escape sequences are special characters with special functionality used in printf() function to format the output according to the user requirement. In the C programming language, we have the following escape sequences...

Escape sequence	Meaning
\n	Moves the cursor to New Line
\t	Inserts Horizontal Tab (5 characters space)
\v	Inserts Vertical Tab (5 lines space)
\a	Beep sound
\b	Backspace (removes the previous character from its current position)
\\	Inserts Backward slash symbol
\?	Inserts Question mark symbol
\'	Inserts Single quotation mark symbol
\"	Inserts Double quotation mark symbol

C Output Functions

Consider the following example program...

Example Program

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

void main(){
   printf("Welcome to\n");
   printf("btechsmartclass\n");
   printf("the perfect website for learning");

}
```

Output:

C Output Functions

putchar() function

The putchar() function is used to display a single character on the output screen. The putchar() functions prints the character which is passed as a parameter to it and returns the same character as a return value. This function is used to print only a single character. To print multiple characters we need to write multiple times or use a looping statement. Consider the following example program...

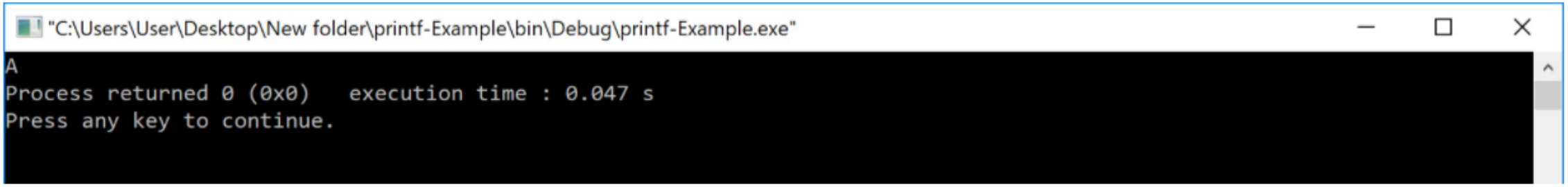
Example Program

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

void main(){
   char ch = 'A';
   putchar(ch);

}
```

Output:



C Output Functions

puts() function

The puts() function is used to display a string on the output screen. The puts() functions prints a string or sequence of characters till the newline.

Consider the following example program...

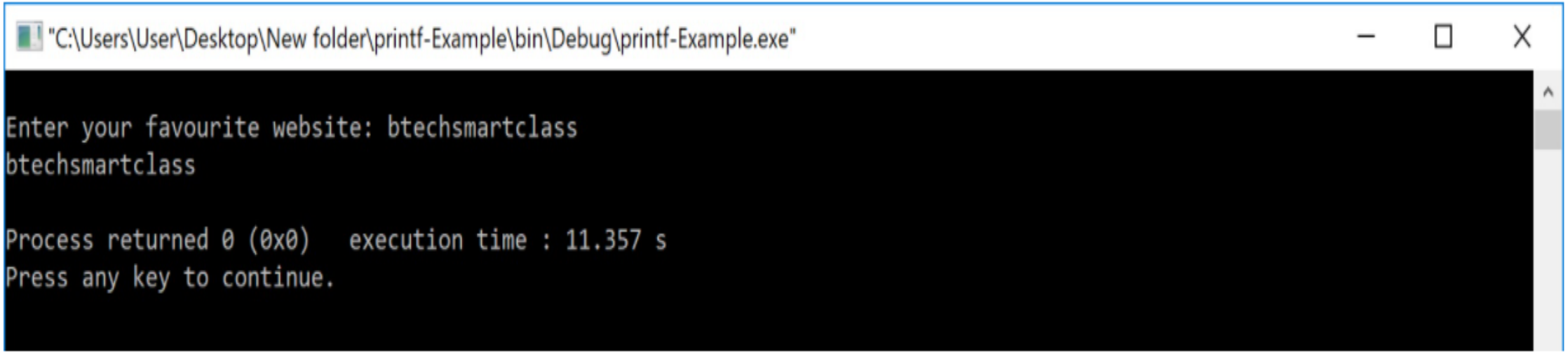
Example Program

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

void main(){
   char name[30];
   printf("\nEnter your favourite website: ");
   gets(name);
   puts(name);
}
```

C Output Functions

Output:



fprintf() function

The fprintf() function is used with the concept of files. The fprintf() function is used to print a line into the file. When you want to use fprintf() function the file must be opened in writting mode.

C Input Functions

C programming language provides built-in functions to perform input operations. The input operations are used to read user values (input) from the keyboard. The c programming language provides the following built-in input functions.

1. scanf()
2. getchar()
3. getch()
4. gets()
5. fscanf()

C Input Functions

scanf() function

The scanf() function is used to read multiple data values of different data types from the keyboard. The scanf() function is built-in function defined in a header file called "**stdio.h**". When we want to use scanf() function in our program, we need to include the respective header file (stdio.h) using **#include** statement. The scanf() function has the following syntax...

Syntax:

```
scanf("format strings",&variableNames);
```

C Input Functions

Example 5: Integer Input/Output

```c
#include <stdio.h>
int main()
{
    int testInteger;
    printf("Enter an integer: ");
    scanf("%d", &testInteger);
    printf("Number = %d",testInteger);
    return 0;
}
```

Output

```
Enter an integer: 4
Number = 4
```

Here, we have used `%d` format specifier inside the `scanf()` function to take `int` input from the user. When the user enters an integer, it is stored in the `testInteger` variable.

> Notice, that we have used `&testInteger` inside `scanf()`. It is because `&testInteger` gets the address of `testInteger`, and the value entered by the user is stored in that address.

C Input Functions

Example 6: Float and Double Input/Output

```c
#include <stdio.h>
int main()
{
    float num1;
    double num2;

    printf("Enter a number: ");
    scanf("%f", &num1);
    printf("Enter another number: ");
    scanf("%lf", &num2);

    printf("num1 = %f\n", num1);
    printf("num2 = %lf", num2);

    return 0;
}
```

Output

```
Enter a number: 12.523
Enter another number: 10.2
num1 = 12.523000
num2 = 10.200000
```

We use `%f` and `%lf` format specifier for `float` and `double` respectively.

C Input Functions

Example 7: C Character I/O

```c
#include <stdio.h>
int main()
{
    char chr;
    printf("Enter a character: ");
    scanf("%c",&chr);
    printf("You entered %c.", chr);
    return 0;
}
```

Output

```
Enter a character: g
You entered g
```

When a character is entered by the user in the above program, the character itself is not stored. Instead, an integer value (ASCII value) is stored.

And when we display that value using `%c` text format, the entered character is displayed. If we use `%d` to display the character, it's ASCII value is printed.

C Input Functions

Example 8: ASCII Value

```c
#include <stdio.h>
int main()
{
    char chr;
    printf("Enter a character: ");
    scanf("%c", &chr);

    // When %c is used, a character is displayed
    printf("You entered %c.\n",chr);

    // When %d is used, ASCII value is displayed
    printf("ASCII value is %d.", chr);
    return 0;
}
```

Output

```
Enter a character: g
You entered g.
ASCII value is 103.
```

C Input Functions

I/O Multiple Values

Here's how you can take multiple inputs from the user and display them.

```c
#include <stdio.h>
int main()
{
    int a;
    float b;

    printf("Enter integer and then a float: ");

    // Taking multiple inputs
    scanf("%d%f", &a, &b);

    printf("You entered %d and %f", a, b);
    return 0;
}
```

Output

```
Enter integer and then a float: -3
3.4
You entered -3 and 3.400000
```

C Input Functions

getchar() function

The getchar() function is used to read a character from the keyboard and return it to the program. This function is used to read a single character. To read multiple characters we need to write multiple times or use a looping statement. Consider the following example program...

Example Program

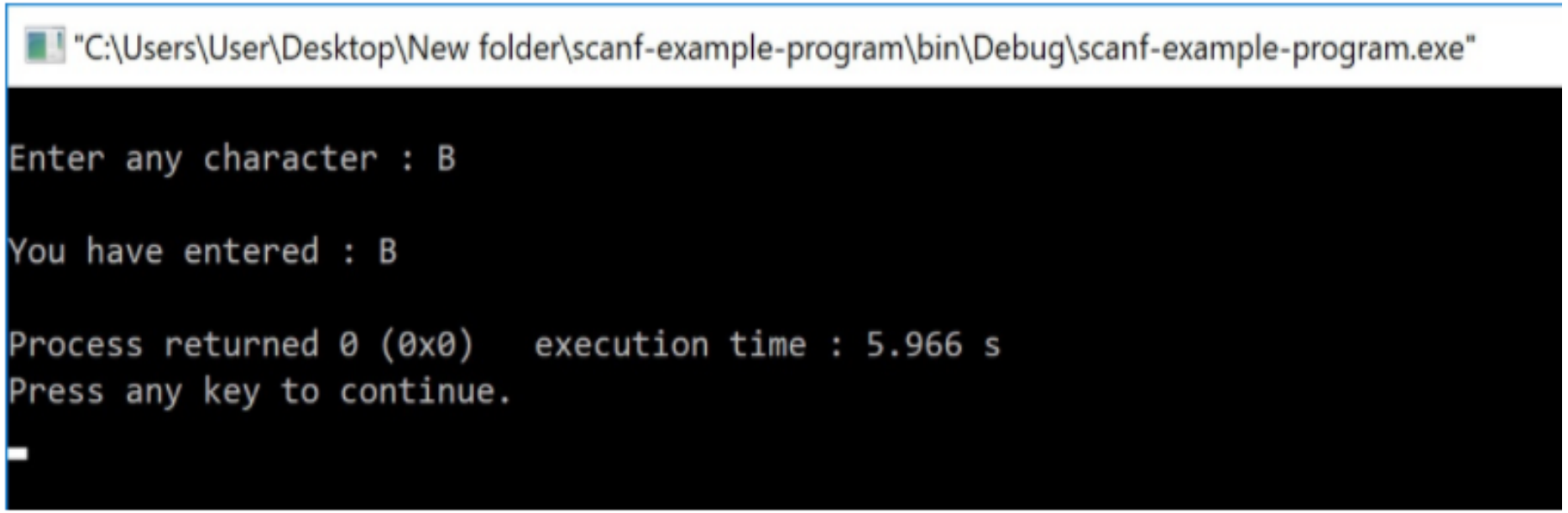
```
#include<stdio.h>
#include<conio.h>

void main(){
   char ch;
   printf("\nEnter any character : ");
   ch = getchar();
   printf("\nYou have entered : %c\n",ch);

}
```

C Input Functions

Output:



C Input Functions

getch() function

The getch() function is similar to getchar function. The getch() function is used to read a character from the keyboard and return it to the program. This function is used to read a single character. To read multiple characters we need to write multiple times or use a looping statement. Consider the following example program...

Example Program

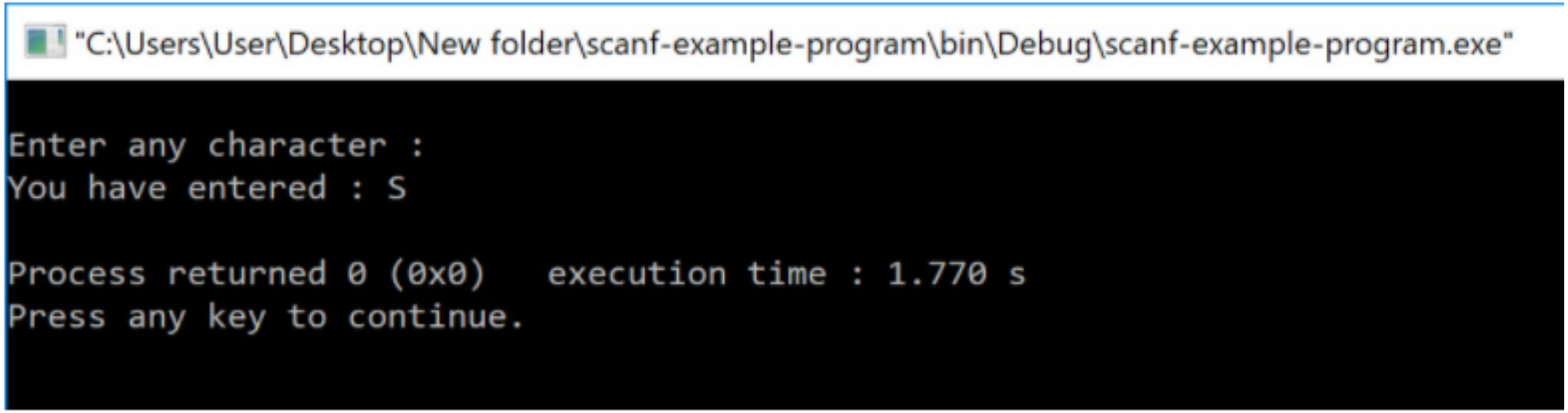
```
#include<stdio.h>
#include<conio.h>

void main(){
   char ch;
   printf("\nEnter any character : ");
   ch = getch();
   printf("\nYou have entered : %c",ch);

}
```

C Input Functions

Output:



C Input Functions

gets() function

The gets() function is used to read a line of string and stores it into a character array. The gets() function reads a line of string or sequence of characters till a newline symbol enters. Consider the following example program...

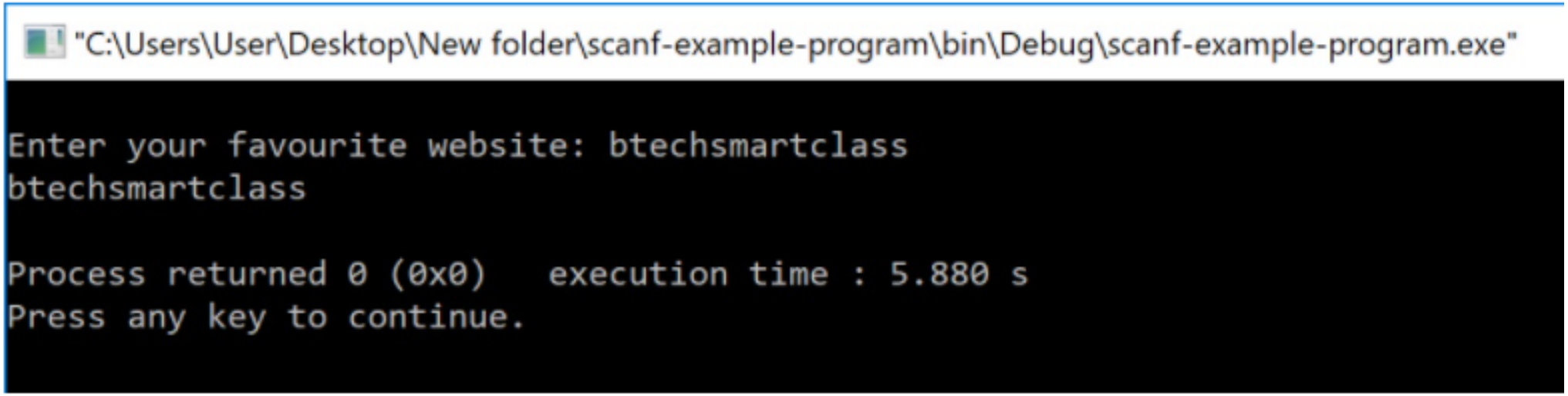
Example Program

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

void main(){
   char name[30];
   printf("\nEnter your favourite website: ");
   gets(name);
   printf("%s",name);
}
```

C Input Functions

Output:



fscanf() function

The fscanf() function is used with the concept of files. The fscanf() function is used to read data values from a file. When you want to use fscanf() function the file must be opened in reading mode.


Thank You!