Basic Information…

1. In every computer system, there is some fixed random access memory.
2. To store the addresses, we need pointers in C programming language.

Declaration of a pointer variable…

1. data_type_whose_address_it_will_store/void, *name_of_variable

Example:

- int *ptr – ptr is a pointer, pointing to an integer means storing the address of an integer variable
- void *ptr - ptr is a void pointer, can contain the address of any data type

2. For finding the address of any variable in the computer memory, we will use “&” operator.

Example:

- Int a = 2, The address of a in memory will be = &a

3. Storing the address of a variable into pointer variable.

Example:

- Int *ptr = &a;

4. To fetch the value which is stored at the memory location, which a pointer has, We will use asterisk “*” operator.

Practical view of pointers...

```c
// The output of this program can be different
// in different runs. Note that the program
// prints address of a variable and a variable
// can be assigned different address in different
// runs.
#include <stdio.h>

int main()
{
    int x;

    // Prints address of x
    printf("%p", &x);

    return 0;
}
```

Variable	x
Address	1000

Practical view of pointers ...

```c
// C program to demonstrate declaration of
// pointer variables.
#include <stdio.h>
int main()
{
    int x = 10;

    // 1) Since there is * in declaration, ptr
    // becomes a pointer varaible (a variable
    // that stores address of another variable)
    // 2) Since there is int before *, ptr is
    // pointer to an integer type variable
    int *ptr;

    // & operator before x is used to get address
    // of x. The address of x is assigned to ptr.
    ptr = &x;

    return 0;
}
```

Variable	x = 10	ptr = 1000			
Address	1000	2000			

Practical view of pointer...(Incrementing of pointers)

```c
// C program to demonstrate use of * for pointers in C
#include <stdio.h>

int main()
{
    // A normal integer variable
    int Var = 10;

    // A pointer variable that holds address of var.
    int *ptr = &Var;

    // This line prints value at address stored in ptr.
    // Value stored is value of variable "var"
    printf("Value of Var = %d\n", *ptr);

    // The output of this line may be different in different
    // runs even on same machine.
    printf("Address of Var = %p\n", ptr);

    // We can also use ptr as lvalue (Left hand
    // side of assignment)
    *ptr = 20; // Value at address is now 20

    // This prints 20
    printf("After doing *ptr = 20, *ptr is %d\n", *ptr);

    return 0;
}
```

Var=10->20	ptr = 1000
1000	2000

***ptr will give the value which is stored at memory location present at ptr pointer variable**

Practical view of pointer...

```
// C++ program to illustrate Pointer Arithmetic
// in C/C++
#include <bits/stdc++.h>

// Driver program
int main()
{
    // Declare an array
    int v[3] = {10, 100, 200};

    // Declare pointer variable
    int *ptr;

    // Assign the address of v[0] to ptr
    ptr = v;

    for (int i = 0; i < 3; i++)
    {
        printf("Value of *ptr = %d\n", *ptr);
        printf("Value of ptr = %p\n\n", ptr);

        // Increment pointer ptr by 1
        ptr++;
    }
}
```

V[0]=10	V[1] = 100	V[2] = 200	ptr = 100
100	104	108	2000

1. *ptr = 10
2. ptr++ means, we are incrementing the address by the size of the data type, which this ptr is pointing to. So ptr will become 104, *ptr→100
3. ptr++ → 108, *(ptr)→200

4

5

6

7

8

Practical view of pointer...

```
// C++ program to illustrate Pointer Arithmetic
// in C/C++
#include <bits/stdc++.h>

// Driver program
int main()
{
    // Declare an array
    int v[3] = {10, 100, 200};

    // Declare pointer variable
    int *ptr;

    // Assign the address of v[0] to ptr
    ptr = v;

    for (int i = 0; i < 3; i++)
    {
        printf("Value of *ptr = %d\n", *ptr);
        printf("Value of ptr = %p\n\n", ptr);

        // Increment pointer ptr by 1
        ptr++;
    }
}
```

V[0]=10	V[1] = 100	V[2] = 200	ptr = 100
100	104	108	2000

1. *ptr = 10
2. ptr++ means, we are incrementing the address by the size of the data type, which this ptr is pointing to. So ptr will become 104, *ptr→100
3. ptr++ → 108, *(ptr)→200

Click to add notes

AutoSave Off | New Microsoft PowerPoint Presentation - Saved to this PC | Vishal Gulia

File Home Insert Design Transitions Animations Slide Show Review View Help Storyboarding Tell me what you want to do | Share | Comments

Paste | Cut | Copy | Format Painter | Clipboard | New Slide | Layout | Reset | Section | Slides | Font | Text Direction | Align Text | Convert to SmartArt | Paragraph | Arrange | Quick Styles | Shape Fill | Shape Outline | Shape Effects | Drawing | Find | Replace | Select | Editing | Dictate | Voice

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp

```
// The output of this program can be different
// in different runs. Note that the program
// prints address of a variable and a variable
// can be assigned different address in different
// runs.
#include <stdio.h>

int main()
{
    int x;

    // Prints address of x
    printf("%p", &x);

    return 0;
}
```

4

5

6

7

8

Practical view of pointer...

```
// C++ program to illustrate Pointer Arithmetic
// in C/C++
#include <bits/stdc++.h>

// Driver program
int main()
{
    // Declare an array
    int v[3] = {10, 100, 200};

    // Declare pointer variable
    int *ptr;

    // Assign the address of v[0] to ptr
    ptr = v;

    for (int i = 0; i < 3; i++)
    {
        printf("Value of *ptr = %d\n", *ptr);
        printf("Value of ptr = %p\n\n", ptr);

        // Increment pointer ptr by 1
        ptr++;
    }
}
```

V[0]=10	V[1] = 100	V[2] = 200	ptr = 100
100	104	108	2000

1. *ptr = 10
2. ptr++ means, we are incrementing the address by the size of the data type, which this ptr is pointing to. So ptr will become 104, *ptr→100
3. ptr++ → 108, *(ptr)→200

Click to add notes

Slide 7 of 8 | Notes | 76%

C++ source file | length : 325 lines : 17 | Ln : 17 Col : 1 Sel : 0 | 0 | Windows (CR LF) UTF-8 | INS | 7:08 PM 3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp basic1.cpp basic2.cpp basic3.cpp

```
1  // The output of this program can be different
2  // in different runs. Note that the program
3  // prints address of a variable and a variable
4  // can be assigned different address in different
5  // runs.
6  #include <stdio.h>
7
8  int main()
9  {
10     int x;
11
12     // Prints address of x
13     printf("%p", &x);
14
15     return 0;
16 }
17
```

C++ source file | length : 325 | lines : 17 | Ln : 17 | Col : 1 | Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:02 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

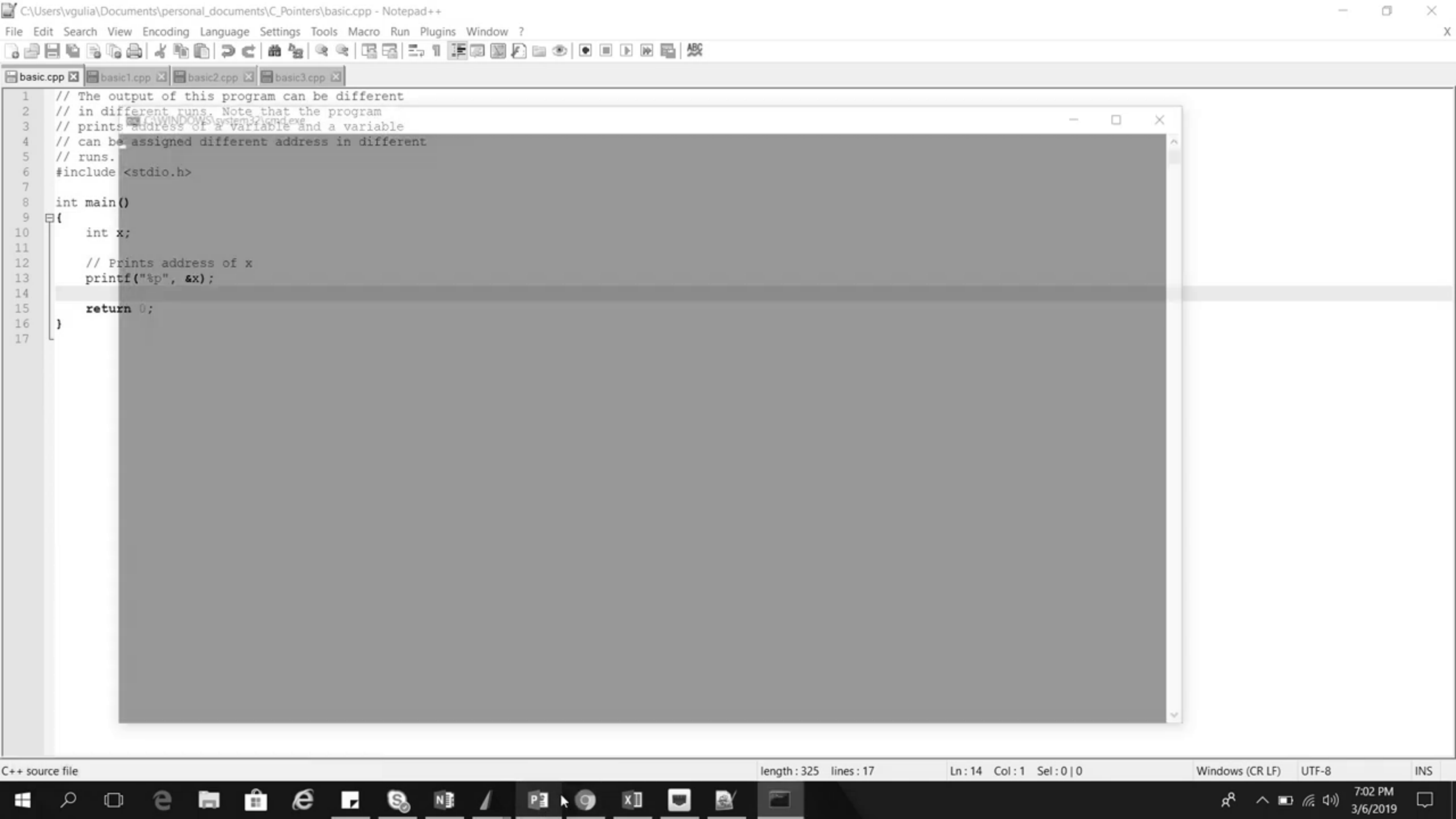
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

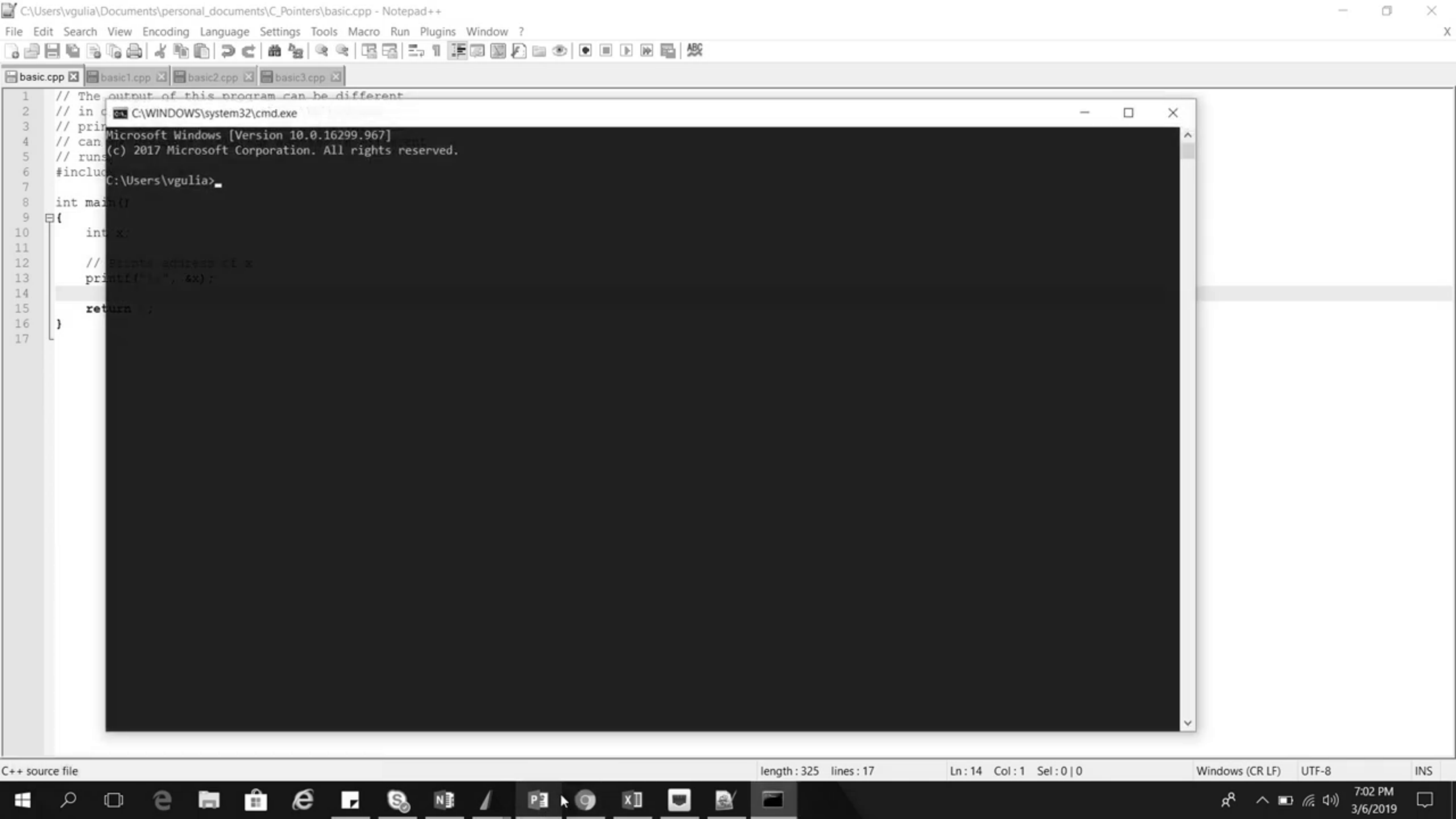
basic.cpp basic1.cpp basic2.cpp basic3.cpp

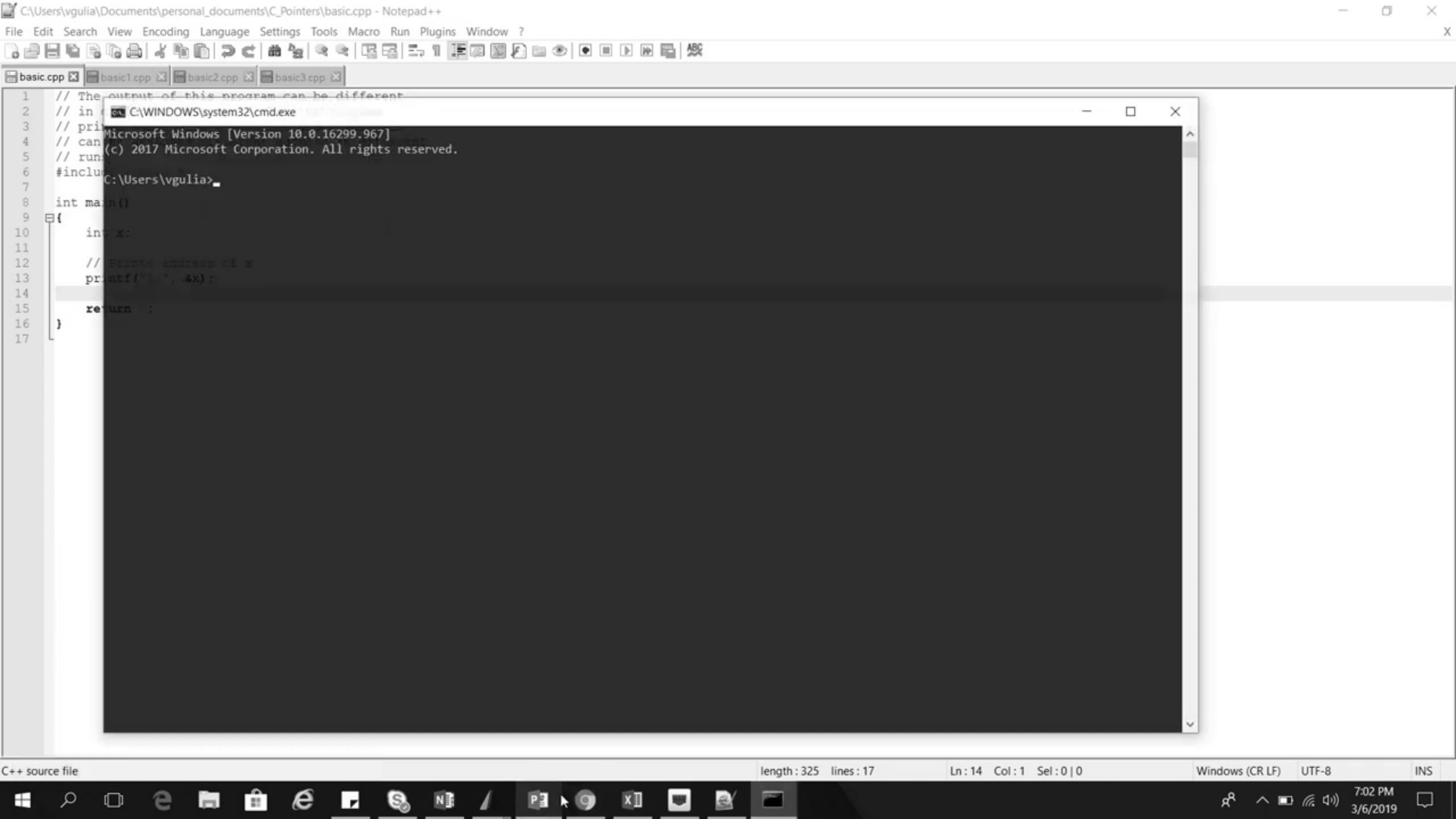
```
// The output of this program can be different
// in different runs. Note that the program
// prints address of a variable and a variable
// can be assigned different address in different
// runs.
#include <stdio.h>

int main()
{
    int x;

    // Prints address of x
    printf("%p", &x);

    return 0;
}
```

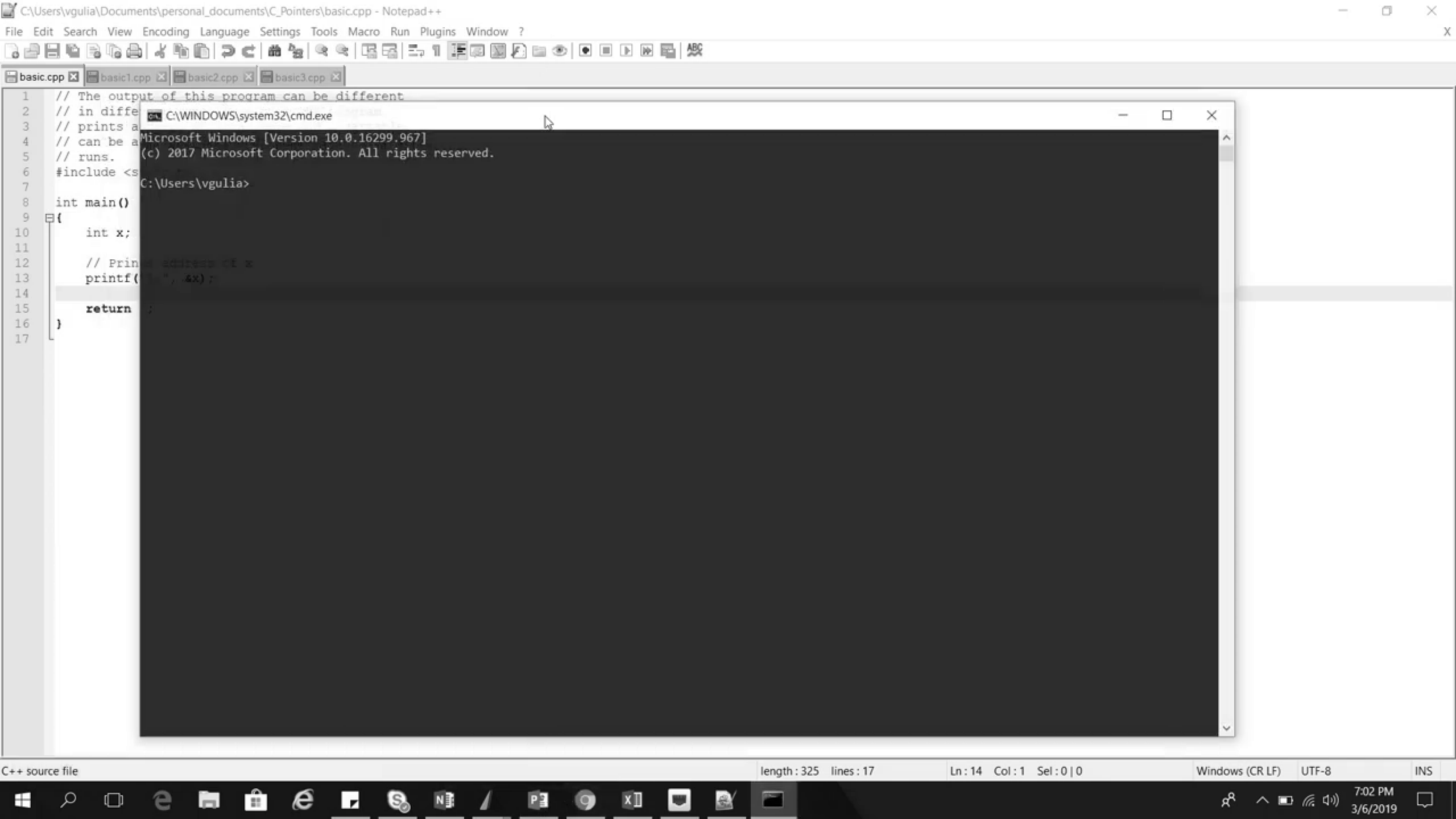
C++ source file | length : 325 | lines : 17 | Ln : 17 | Col : 1 | Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

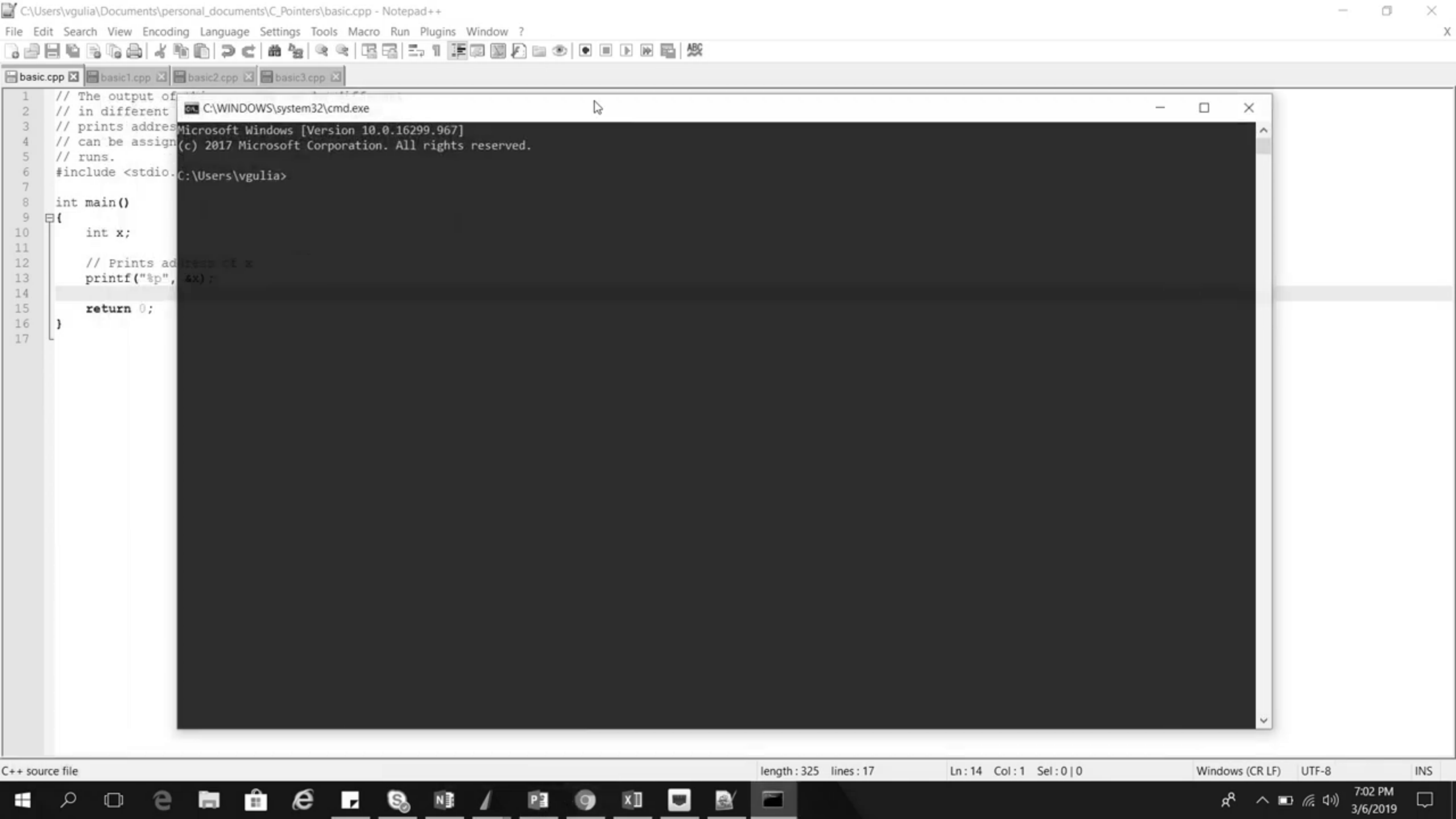
7:02 PM
3/6/2019

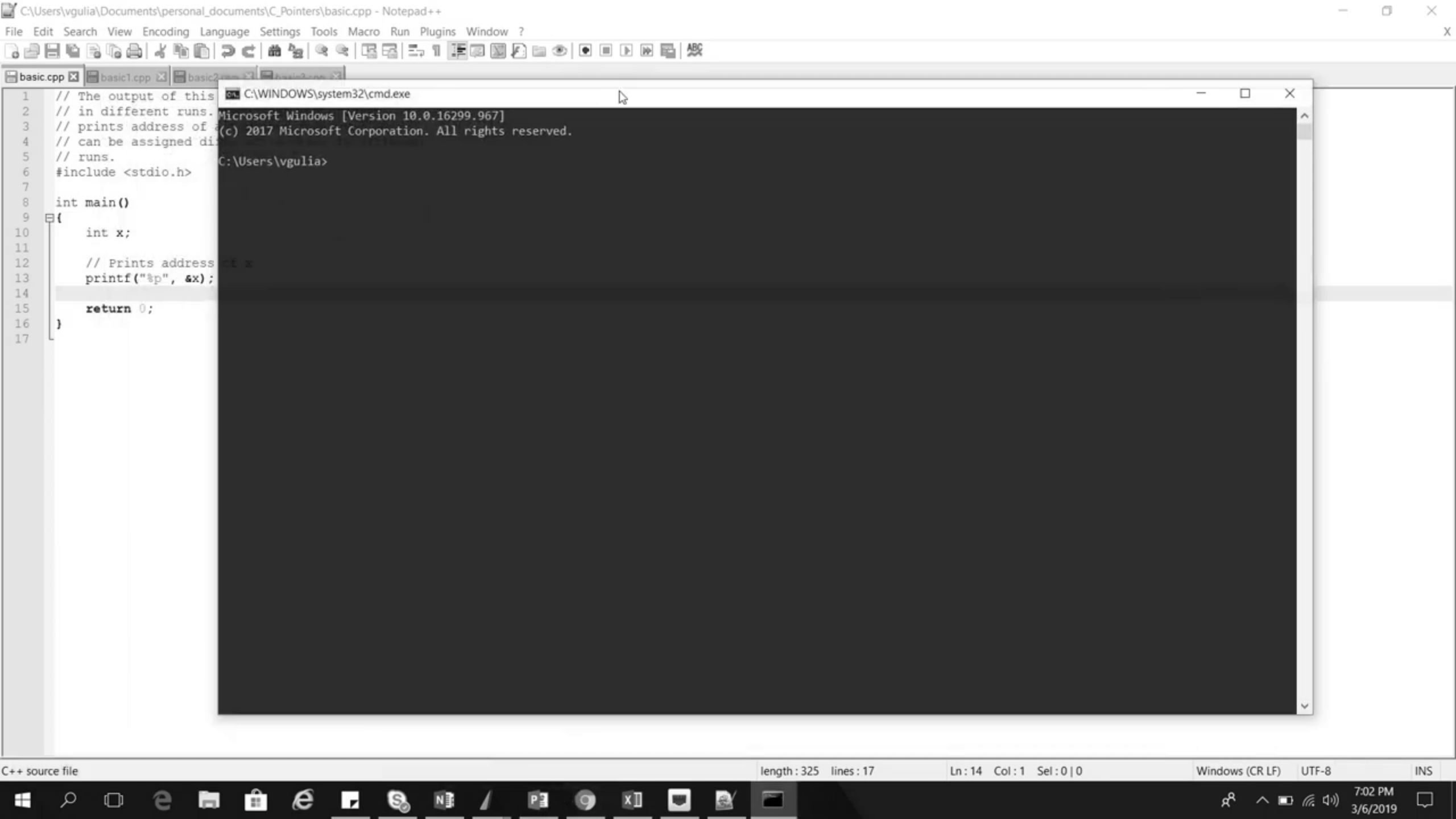

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
basic3.cpp
// The output of this program can be different
// in different runs. Note that the program
// prints address of a variable and a variable
// can be assigned different address in different
// runs.
#include <stdio.h>
int main()
{
int x;
// Prints address of x
printf("%p", &x);
return 0;
}
C:\WINDOWS\system32\cmd.exe
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:02 PM
3/6/2019

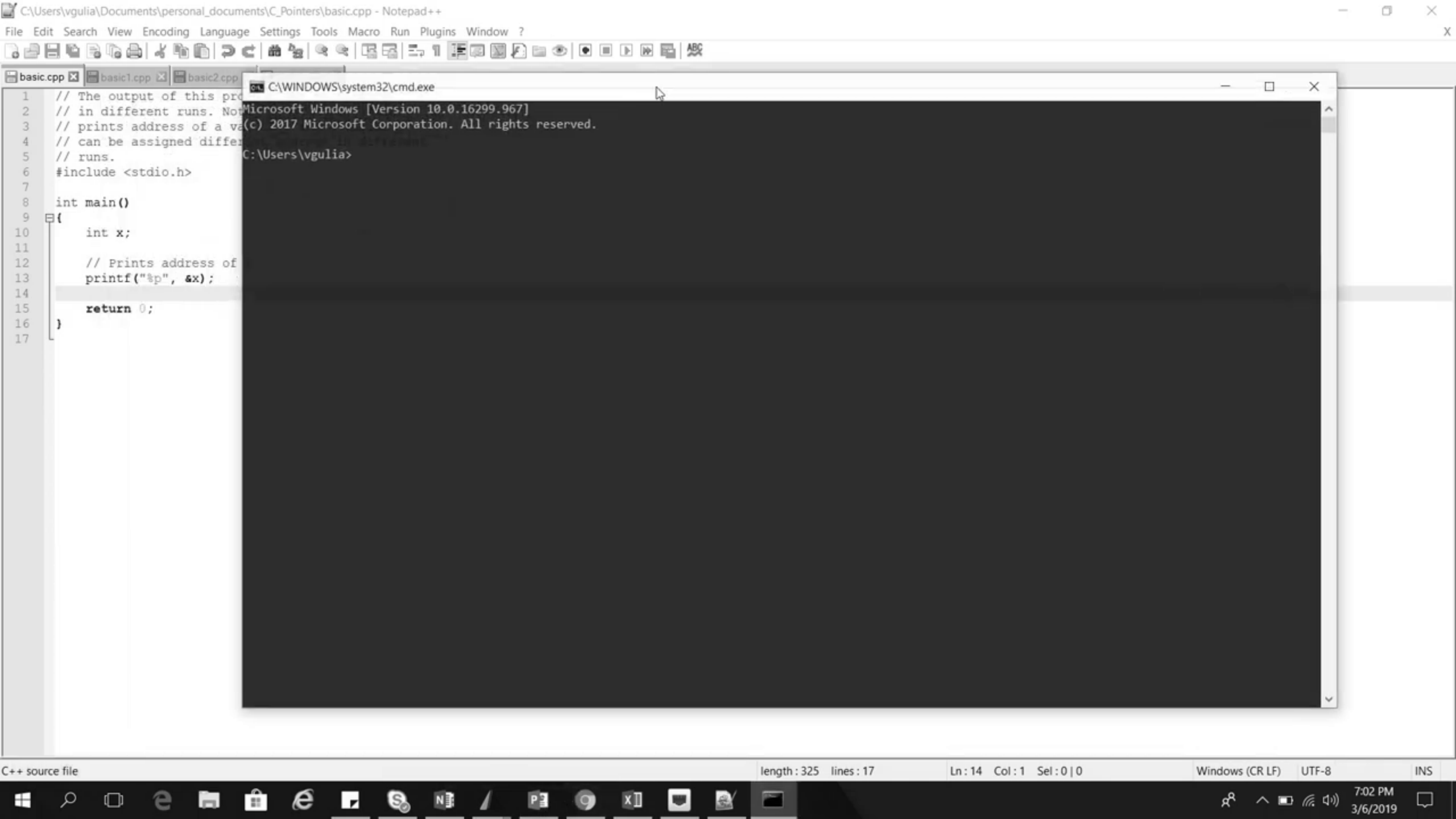

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
basic3.cpp
// The output of this program can be different
// in c
// prin
// can
// runs
#includ
int main()
{
int x;
// Prints address of x
printf("%p", &x);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:02 PM
3/6/2019

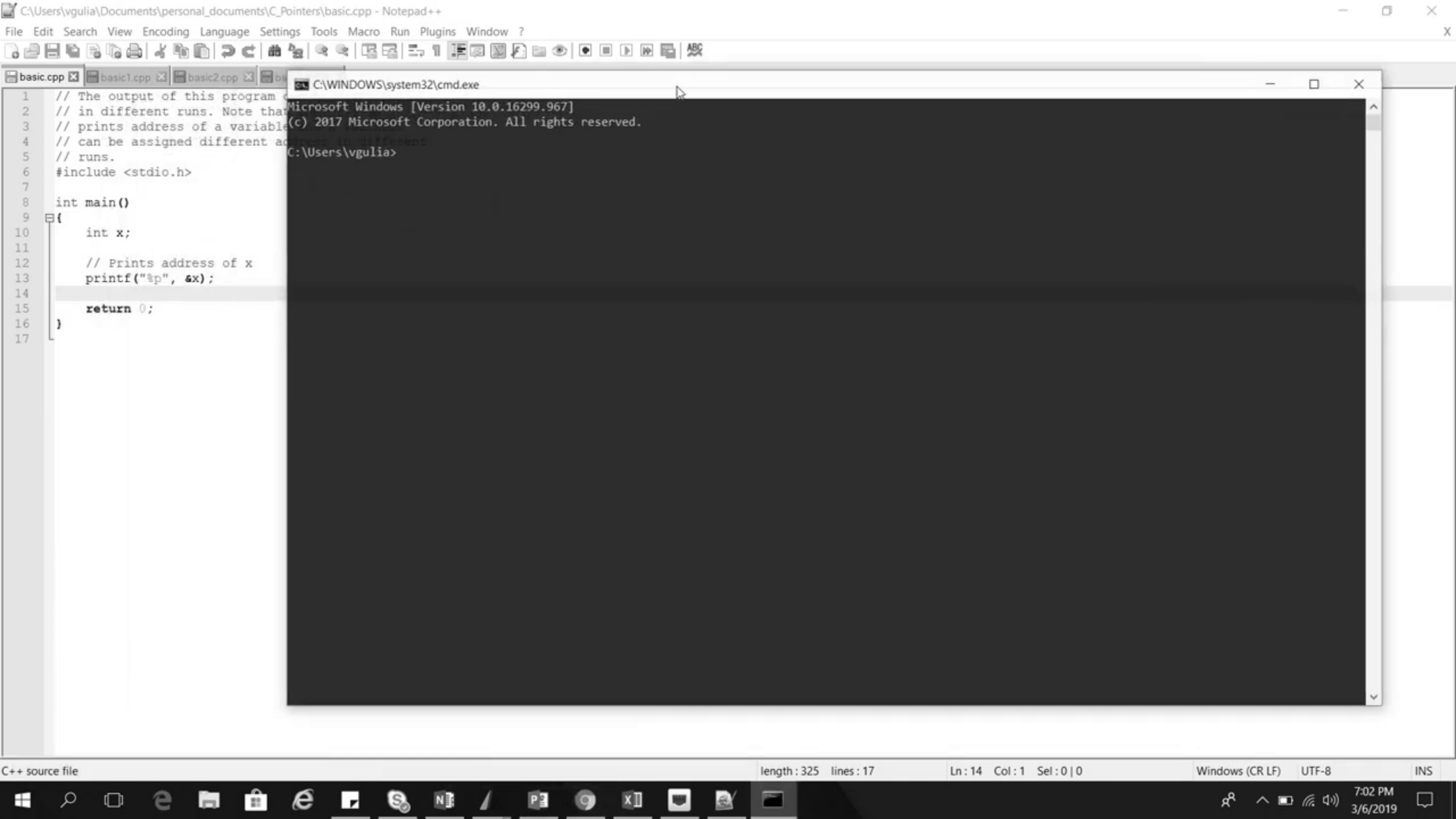

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
basic3.cpp
// The output of this program can be different
// in
// pri
// can
// run
#inclu
int ma n()
{
in x;
// Prints address of x
pr ntf("%p", &x);
re urn ;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:02 PM
3/6/2019

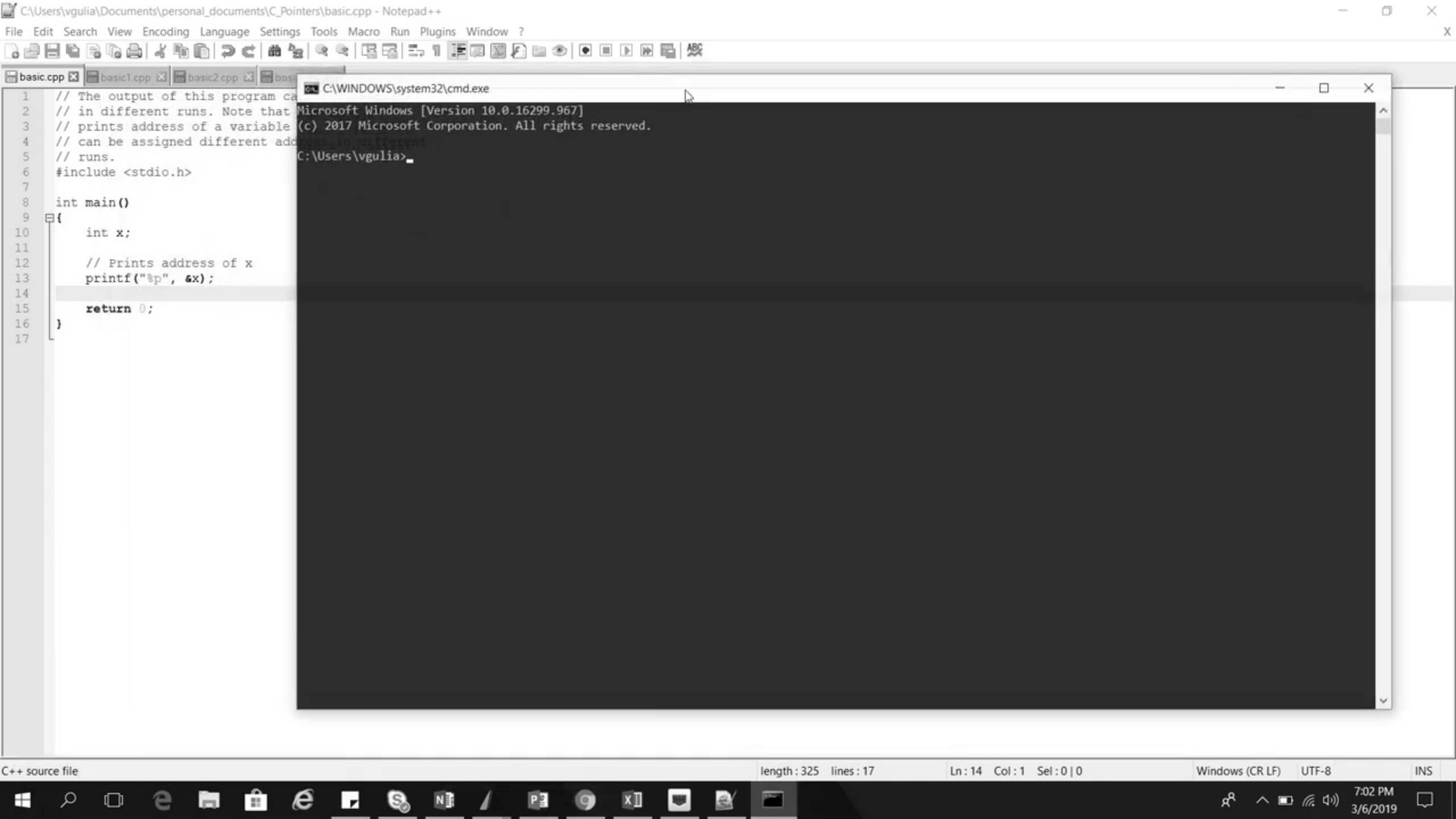

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
basic3.cpp
// The output of this program can be different
// in diffe
// prints a
// can be a
// runs.
#include <s
int main()
{
int x;
// Prin address of x
printf(", &x);
return ;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:02 PM
3/6/2019

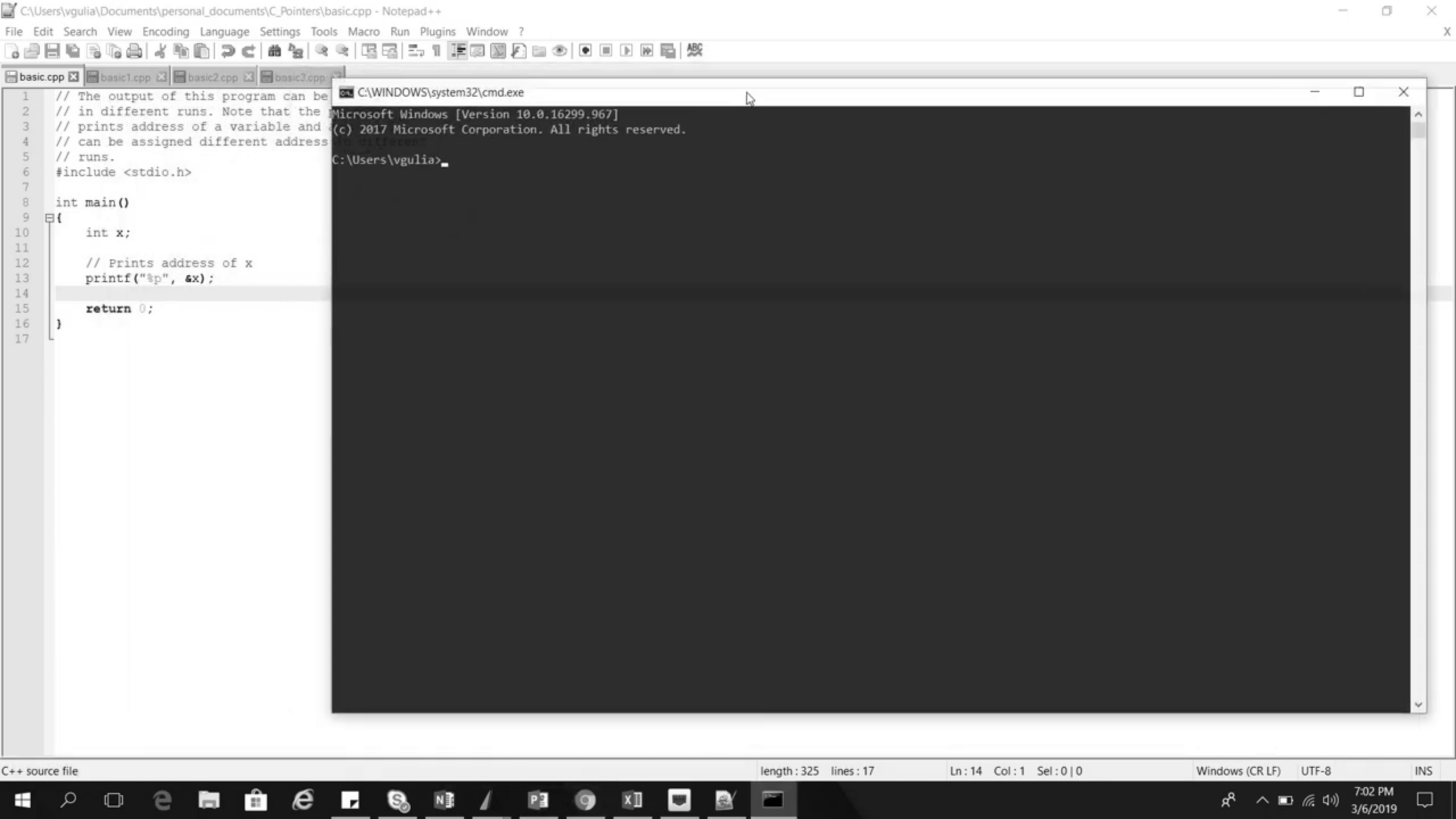

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
basic3.cpp
// The output of
// in different
// prints addres
// can be assign
// runs.
#include <stdio.
int main()
{
int x;
// Prints ad
printf("%p", &x);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:02 PM
3/6/2019

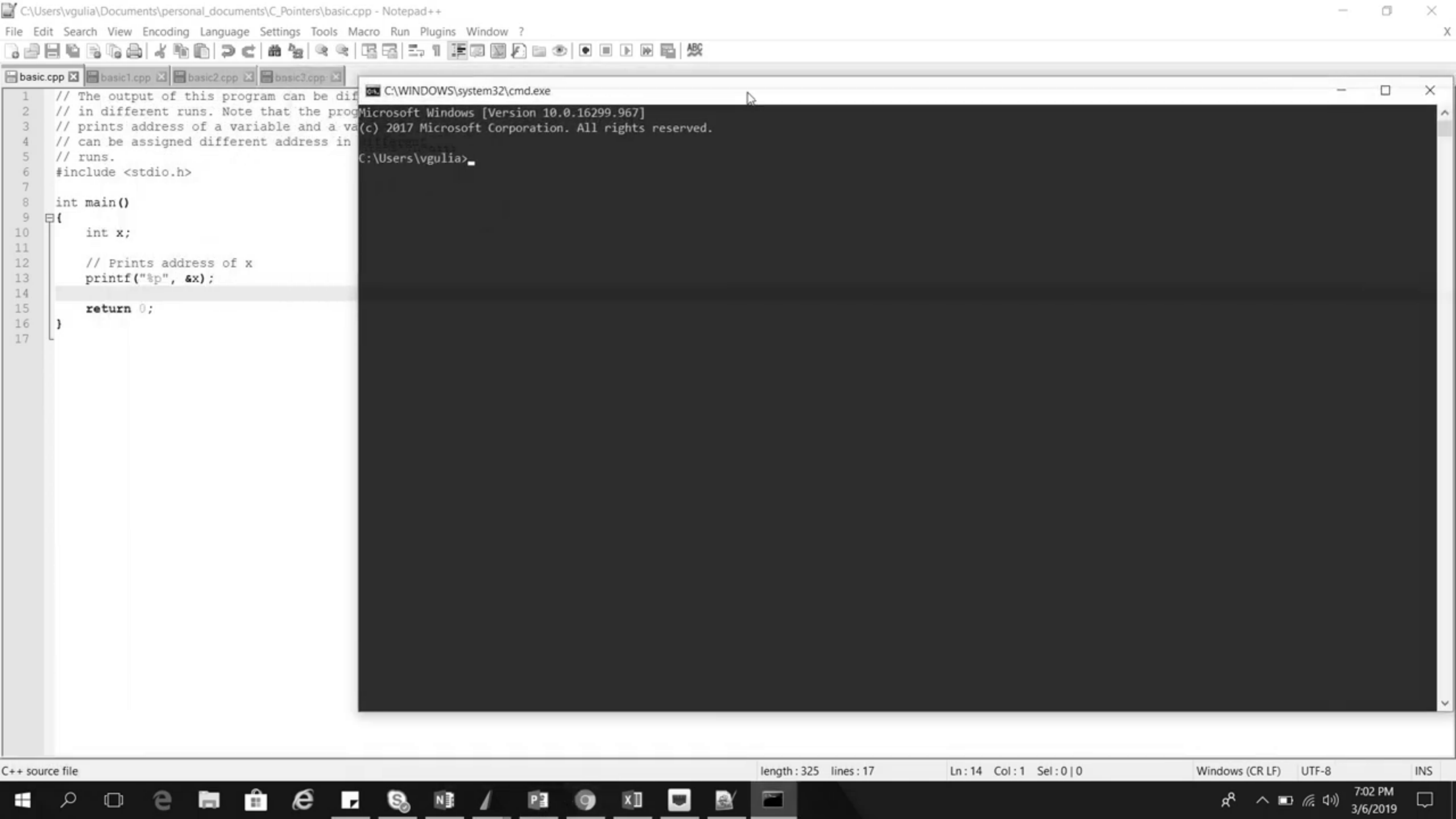

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
// The output of this
// in different runs.
// prints address of
// can be assigned di
// runs.
#include <stdio.h>
int main()
{
int x;
// Prints address of x
printf("%p", &x);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:02 PM
3/6/2019


C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
// The output of this pr
// in different runs. No
// prints address of a va
// can be assigned differ
// runs.
#include <stdio.h>
int main()
{
int x;
// Prints address of
printf("%p", &x);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:02 PM
3/6/2019


C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
// The output of this program
// in different runs. Note that
// prints address of a variable
// can be assigned different addresses in different
// runs.
#include <stdio.h>
int main()
{
int x;
// Prints address of x
printf("%p", &x);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:02 PM
3/6/2019


C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
// The output of this program ca
// in different runs. Note that
// prints address of a variable
// can be assigned different add
// runs.
#include <stdio.h>
int main()
{
int x;
// Prints address of x
printf("%p", &x);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:02 PM
3/6/2019


C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
basic3.cpp
// The output of this program can be
// in different runs. Note that the
// prints address of a variable and
// can be assigned different address
// runs.
#include <stdio.h>
int main()
{
int x;
// Prints address of x
printf("%p", &x);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF) UTF-8 INS
7:02 PM
3/6/2019


C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
basic3.cpp
// The output of this program can be dif
// in different runs. Note that the prog
// prints address of a variable and a va
// can be assigned different address in
// runs.
#include <stdio.h>
int main()
{
int x;
// Prints address of x
printf("%p", &x);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF) UTF-8 INS
7:02 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

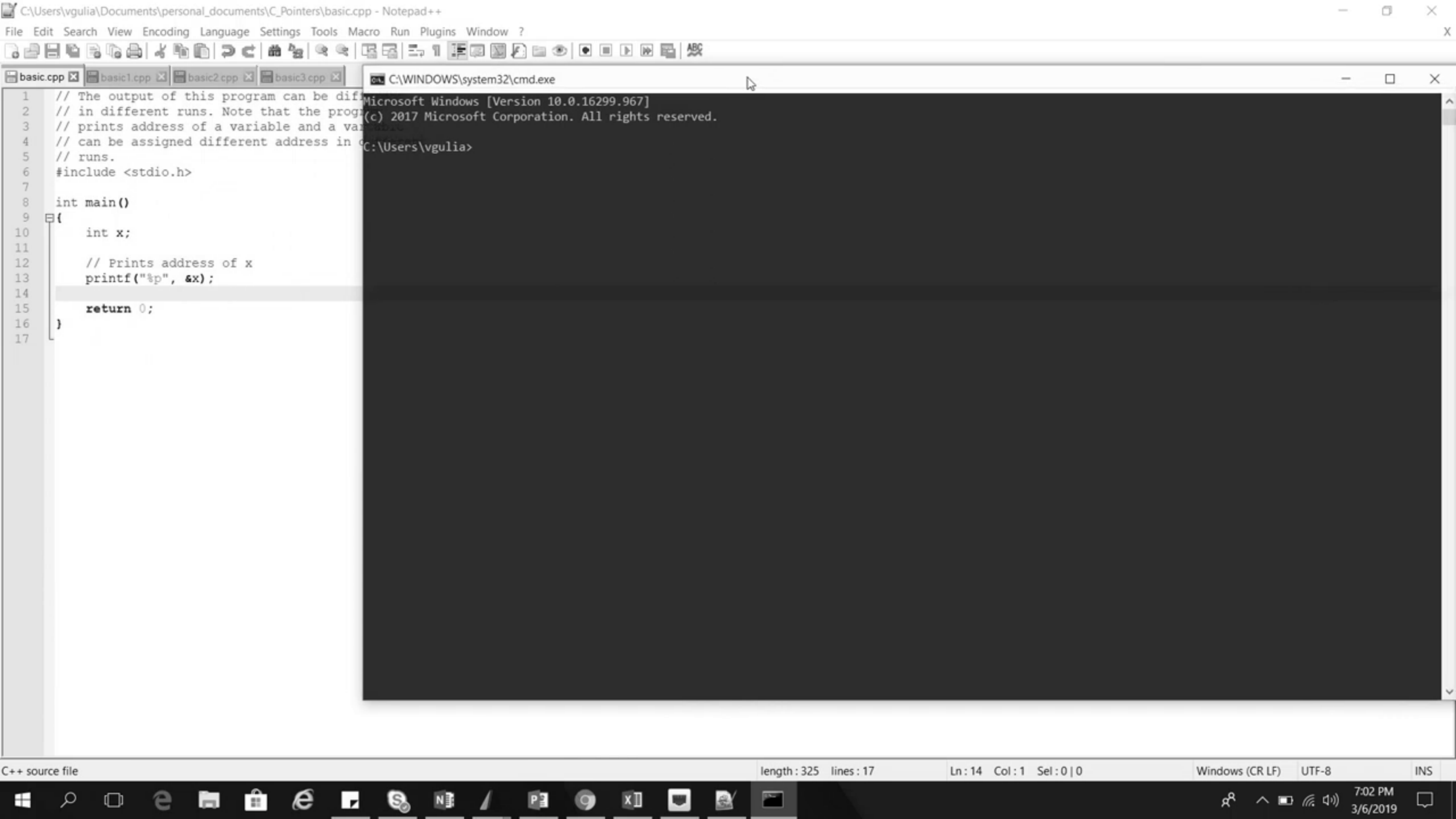
```
1  // The output of this program can be diff
2  // in different runs. Note that the progr
3  // prints address of a variable and a var
4  // can be assigned different address in d
5  // runs.
6  #include <stdio.h>
7
8  int main()
9  {
10     int x;
11
12     // Prints address of x
13     printf("%p", &x);
14
15     return 0;
16 }
17
```

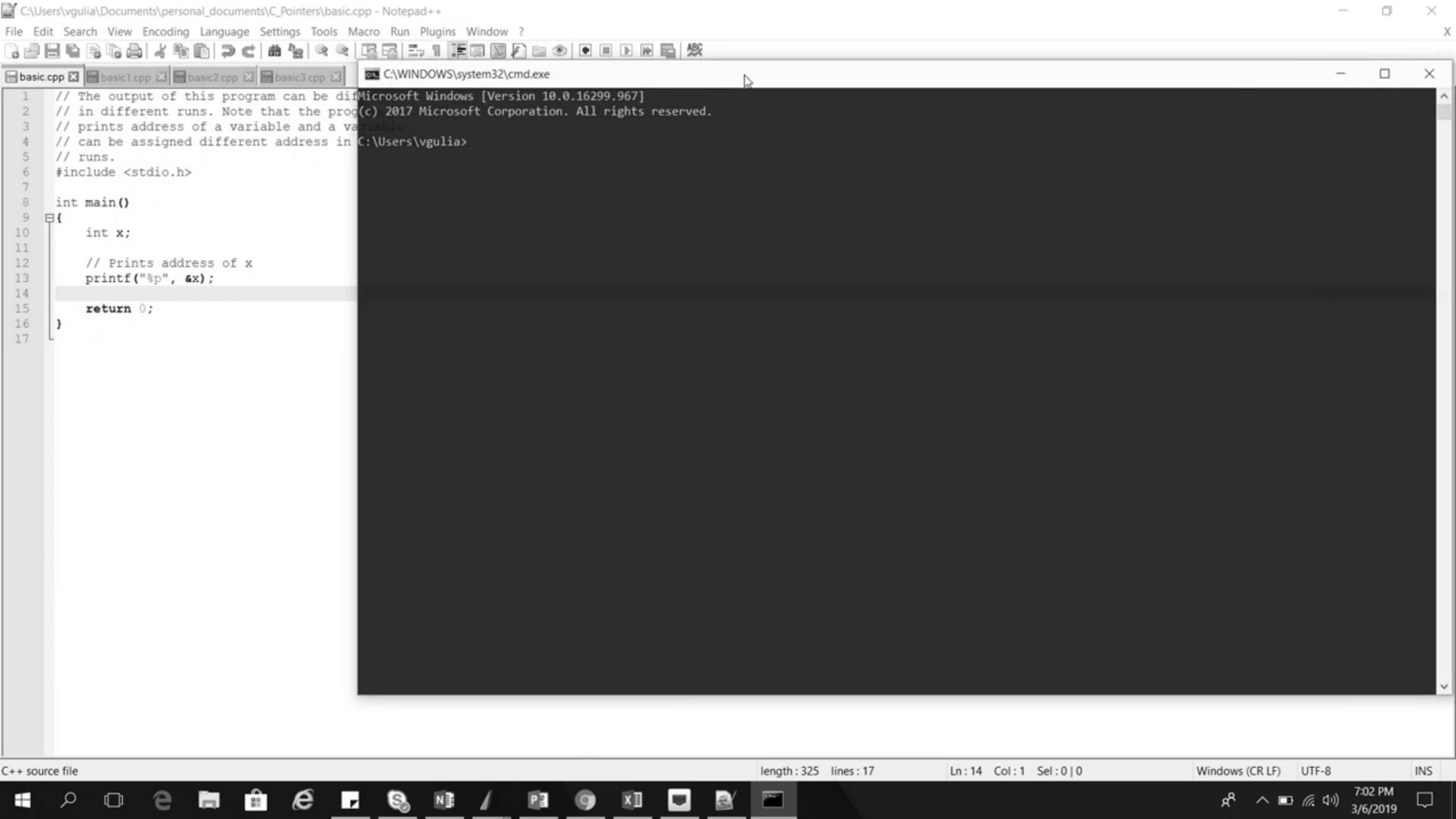
C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>
```

C++ source file | length : 325 lines : 17 | Ln : 14 Col : 1 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:02 PM
3/6/2019


C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
basic3.cpp
// The output of this program can be dif
// in different runs. Note that the prog
// prints address of a variable and a va
// can be assigned different address in
// runs.
#include <stdio.h>
int main()
{
int x;
// Prints address of x
printf("%p", &x);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:02 PM
3/6/2019


C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
basic3.cpp
// The output of this program can be dif
// in different runs. Note that the prog
// prints address of a variable and a va
// can be assigned different address in
// runs.
#include <stdio.h>
int main()
{
int x;
// Prints address of x
printf("%p", &x);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>
C++ source file
length : 325 lines : 17
Ln : 14 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:02 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp basic1.cpp basic2.cpp basic3.cpp

```
1  // The output of this program can be different
2  // in different runs. Note that the program
3  // prints address of a variable and a variable
4  // can be assigned different address in different
5  // runs.
6  #include <stdio.h>
7
8  int main()
9  {
10     int x;
11
12     // Prints address of x
13     printf("%p", &x);
14
15     return 0;
16 }
17
```

C++ source file | length : 325 | lines : 17 | Ln : 6 | Col : 20 | Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic1.cpp - Notepad++

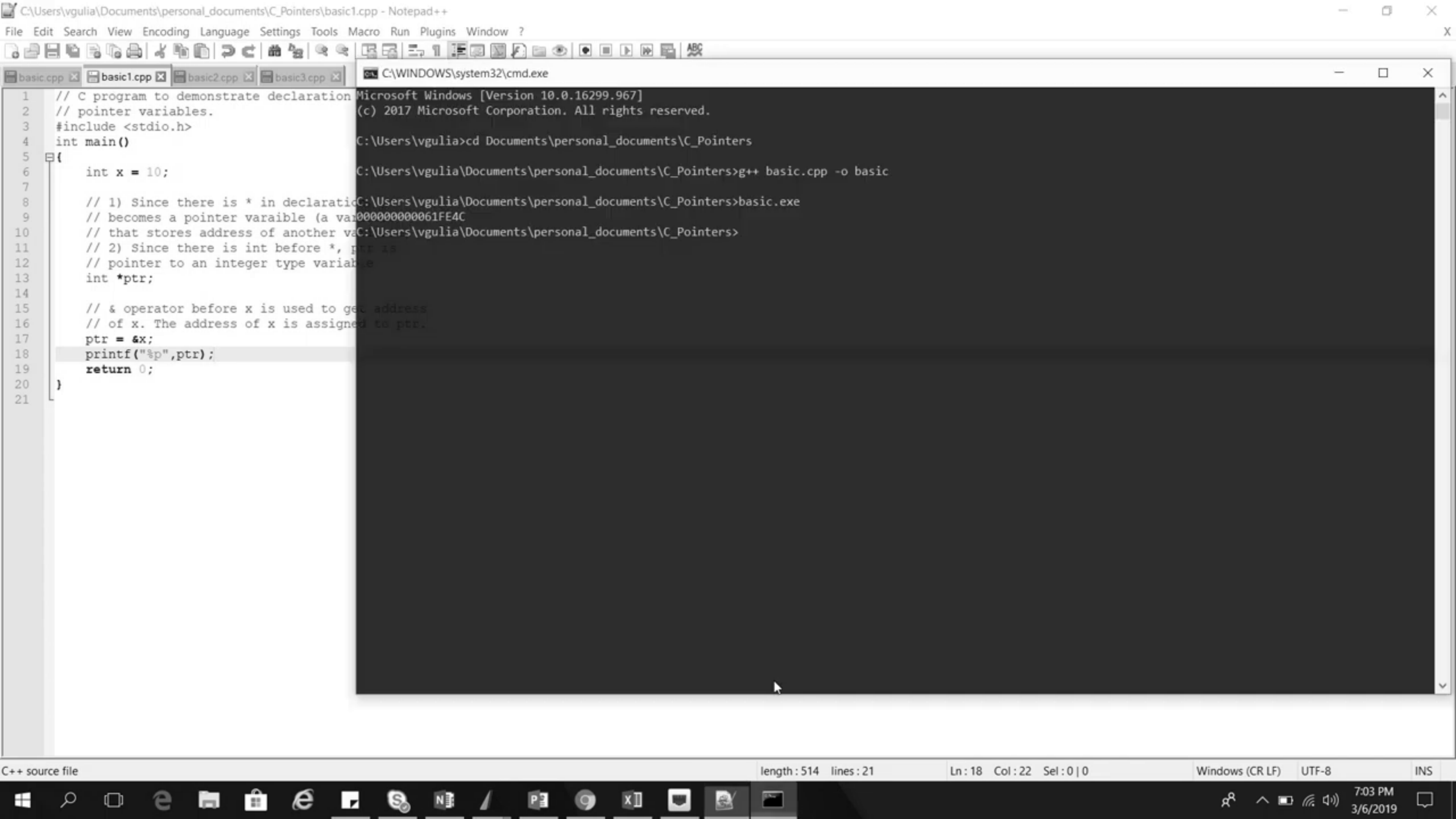
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

```
1  // C program to demonstrate declaration of
2  // pointer variables.
3  #include <stdio.h>
4  int main()
5  {
6      int x = 10;
7
8      // 1) Since there is * in declaration, ptr
9      // becomes a pointer varaible (a variable
10     // that stores address of another variable)
11     // 2) Since there is int before *, ptr is
12     // pointer to an integer type variable
13     int *ptr;
14
15     // & operator before x is used to get address
16     // of x. The address of x is assigned to ptr.
17     ptr = &x;
18     printf("%p",ptr);
19     return 0;
20 }
21
```

C++ source file | length : 514 | lines : 21 | Ln : 18 | Col : 22 | Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM
3/6/2019


C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic1.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp
basic1.cpp
basic2.cpp
basic3.cpp
// C program to demonstrate declaration
// pointer variables.
#include <stdio.h>
int main()
{
int x = 10;
// 1) Since there is * in declaratic
// becomes a pointer varaible (a va
// that stores address of another va
// 2) Since there is int before *, ptr is
// pointer to an integer type variable
int *ptr;
// & operator before x is used to get address
// of x. The address of x is assigned to ptr.
ptr = &x;
printf("%p",ptr);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>cd Documents\personal_documents\C_Pointers
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp -o basic
C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic.exe
000000000061FE4C
C:\Users\vgulia\Documents\personal_documents\C_Pointers>
C++ source file
length : 514 lines : 21
Ln : 18 Col : 22 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:03 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic1.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

```
// C program to demonstrate declaration of
// pointer variables.
#include <stdio.h>
int main()
{
    int x = 10;

    // 1) Since there is * in declaration, ptr
    // becomes a pointer varaible (a variable
    // that stores address of another variable)
    // 2) Since there is int before *, ptr is
    // pointer to an integer type variable
    int *ptr;

    // & operator before x is used to get address
    // of x. The address of x is assigned to ptr.
    ptr = &x;
    printf("%p",ptr);
    return 0;
}
```

C++ source file | length : 514 lines : 21 | Ln : 12 Col : 20 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic2.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

```
1  // C program to demonstrate use of * for pointers in C
2  #include <stdio.h>
3
4  int main()
5  {
6      // A normal integer variable
7      int Var = 10;
8
9      // A pointer variable that holds address of var.
10     int *ptr = &Var;
11
12     // This line prints value at address stored in ptr.
13     // Value stored is value of variable "var"
14     printf("Value of Var = %d\n", *ptr);
15
16     // The output of this line may be different in different
17     // runs even on same machine.
18     printf("Address of Var = %p\n", ptr);
19
20     // We can also use ptr as lvalue (Left hand
21     // side of assignment)
22     *ptr = 20; // Value at address is now 20
23
24     // This prints 20
25     printf("After doing *ptr = 20, *ptr is %d\n", *ptr);
26
27     return 0;
28 }
29
```

C++ source file | length : 716 | lines : 29 | Ln : 29 | Col : 1 | Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic2.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

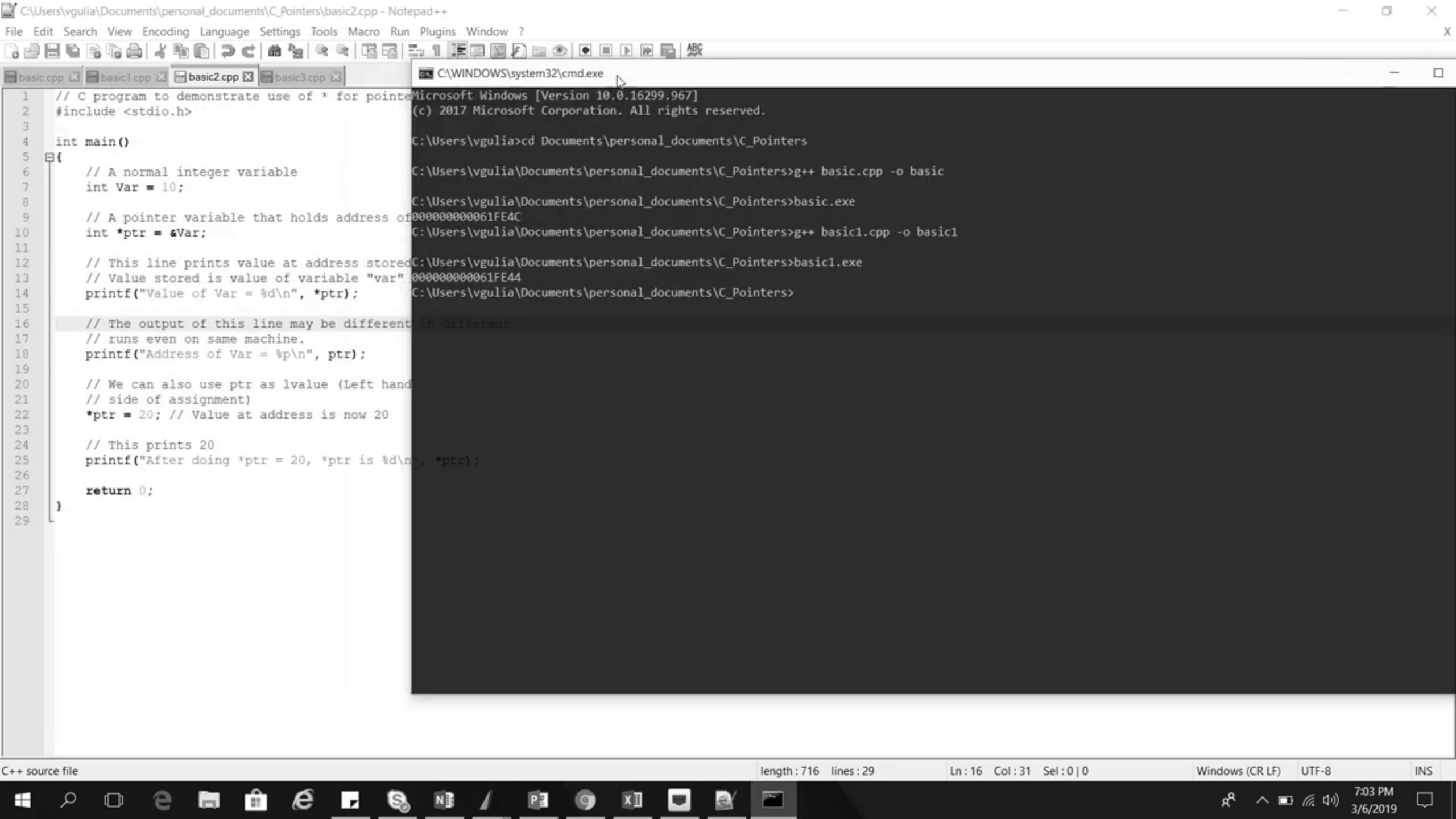
```
1  // C program to demonstrate use of * for
2  #include <stdio.h>
3
4  int main()
5  {
6      // A normal integer variable
7      int Var = 10;
8
9      // A pointer variable that holds add
10     int *ptr = &Var;
11
12     // This line prints value at address
13     // Value stored is value of variable
14     printf("Value of Var = %d\n", *ptr);
15
16     // The output of this line may be di
17     // runs even on same machine.
18     printf("Address of Var = %p\n", ptr);
19
20     // We can also use ptr as lvalue (Le
21     // side of assignment)
22     *ptr = 20; // Value at address is no
23
24     // This prints 20
25     printf("After doing *ptr = 20, *ptr
26
27     return 0;
28 }
29
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents\personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp -o basic

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic.exe
000000000061FE4C
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic1.cpp -o basic1

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic1.exe
000000000061FE44
C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C++ source file | length : 716 | lines : 29 | Ln : 16 | Col : 31 | Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM
3/6/2019


C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic2.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
basic.cpp basic1.cpp basic2.cpp basic3.cpp
// C program to demonstrate use of * for pointe
#include <stdio.h>
int main()
{
// A normal integer variable
int Var = 10;
// A pointer variable that holds address of
int *ptr = &Var;
// This line prints value at address stored
// Value stored is value of variable "var"
printf("Value of Var = %d\n", *ptr);
// The output of this line may be different in different
// runs even on same machine.
printf("Address of Var = %p\n", ptr);
// We can also use ptr as lvalue (Left hand
// side of assignment)
*ptr = 20; // Value at address is now 20
// This prints 20
printf("After doing *ptr = 20, *ptr is %d\n", *ptr);
return 0;
}
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.
C:\Users\vgulia>cd Documents\personal_documents\C_Pointers
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp -o basic
C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic.exe
000000000061FE4C
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic1.cpp -o basic1
C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic1.exe
000000000061FE44
C:\Users\vgulia\Documents\personal_documents\C_Pointers>
C++ source file
length : 716 lines : 29
Ln : 16 Col : 31 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS
7:03 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic2.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

```
1  // C program to demonstrate use of * for pointers
2  #include <stdio.h>
3
4  int main()
5  {
6      // A normal integer variable
7      int Var = 10;
8
9      // A pointer variable that holds address of va
10     int *ptr = &Var;
11
12     // This line prints value at address stored in
13     // Value stored is value of variable "var"
14     printf("Value of Var = %d\n", *ptr);
15
16     // The output of this line may be different in different
17     // runs even on same machine.
18     printf("Address of Var = %p\n", ptr);
19
20     // We can also use ptr as lvalue (Left hand
21     // side of assignment)
22     *ptr = 20; // Value at address is now 20
23
24     // This prints 20
25     printf("After doing *ptr = 20, *ptr is %d\n", *ptr);
26
27     return 0;
28 }
29
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents\personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp -o basic

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic.exe
000000000061FE4C
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic1.cpp -o basic1

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic1.exe
000000000061FE44
C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C++ source file | length : 716 lines : 29 | Ln : 16 Col : 31 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic2.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

```
// C program to demonstrate use of * for pointers
#include <stdio.h>

int main()
{
    // A normal integer variable
    int Var = 10;

    // A pointer variable that holds address of va
    int *ptr = &Var;

    // This line prints value at address stored in
    // Value stored is value of variable "var"
    printf("Value of Var = %d\n", *ptr);

    // The output of this line may be different in different
    // runs even on same machine.
    printf("Address of Var = %p\n", ptr);

    // We can also use ptr as lvalue (Left hand
    // side of assignment)
    *ptr = 20; // Value at address is now 20

    // This prints 20
    printf("After doing *ptr = 20, *ptr is %d\n", *ptr);

    return 0;
}
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents\personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp -o basic

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic.exe
000000000061FE4C
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic1.cpp -o basic1

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic1.exe
000000000061FE44
C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C++ source file | length : 716 lines : 29 | Ln : 16 Col : 31 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic2.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

```
// C program to demonstrate use of * for pointers in
#include <stdio.h>

int main()
{
    // A normal integer variable
    int Var = 10;

    // A pointer variable that holds address of var.
    int *ptr = &Var;

    // This line prints value at address stored in p
    // Value stored is value of variable "var"
    printf("Value of Var = %d\n", *ptr);

    // The output of this line may be different in different
    // runs even on same machine.
    printf("Address of Var = %p\n", ptr);

    // We can also use ptr as lvalue (Left hand
    // side of assignment)
    *ptr = 20; // Value at address is now 20

    // This prints 20
    printf("After doing *ptr = 20, *ptr is %d\n", *ptr);

    return 0;
}
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents\personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp -o basic

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic.exe
000000000061FE4C
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic1.cpp -o basic1

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic1.exe
000000000061FE44
C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C++ source file | length : 716 lines : 29 | Ln : 16 Col : 31 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic2.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

```
1  // C program to demonstrate use of * for pointers in C
2  #include <stdio.h>
3
4  int main()
5  {
6      // A normal integer variable
7      int Var = 10;
8
9      // A pointer variable that holds address of var.
10     int *ptr = &Var;
11
12     // This line prints value at address stored in ptr
13     // Value stored is value of variable "var"
14     printf("Value of Var = %d\n", *ptr);
15
16     // The output of this line may be different in different
17     // runs even on same machine.
18     printf("Address of Var = %p\n", ptr);
19
20     // We can also use ptr as lvalue (Left hand
21     // side of assignment)
22     *ptr = 20; // Value at address is now 20
23
24     // This prints 20
25     printf("After doing *ptr = 20, *ptr is %d\n", *ptr);
26
27     return 0;
28 }
29
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents\personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp -o basic

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic.exe
000000000061FE4C
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic1.cpp -o basic1

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic1.exe
000000000061FE44
C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C++ source file | length : 716 lines : 29 | Ln : 16 Col : 31 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic2.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

```
// C program to demonstrate use of * for pointers in C
#include <stdio.h>

int main()
{
    // A normal integer variable
    int Var = 10;

    // A pointer variable that holds address of var.
    int *ptr = &Var;

    // This line prints value at address stored in ptr.
    // Value stored is value of variable "var"
    printf("Value of Var = %d\n", *ptr);

    // The output of this line may be different in different
    // runs even on same machine.
    printf("Address of Var = %p\n", ptr);

    // We can also use ptr as lvalue (Left hand
    // side of assignment)
    *ptr = 20; // Value at address is now 20

    // This prints 20
    printf("After doing *ptr = 20, *ptr is %d\n", *ptr);

    return 0;
}
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents\personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp -o basic

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic.exe
000000000061FE4C
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic1.cpp -o basic1

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic1.exe
000000000061FE44
C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C++ source file | length : 716 lines : 29 | Ln : 16 Col : 31 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM 3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic2.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

```
1  // C program to demonstrate use of * for pointers in C
2  #include <stdio.h>
3
4  int main()
5  {
6      // A normal integer variable
7      int Var = 10;
8
9      // A pointer variable that holds address of var.
10     int *ptr = &Var;
11
12     // This line prints value at address stored in ptr.
13     // Value stored is value of variable "var"
14     printf("Value of Var = %d\n", *ptr);
15
16     // The output of this line may be different in diffe
17     // runs even on same machine.
18     printf("Address of Var = %p\n", ptr);
19
20     // We can also use ptr as lvalue (Left hand
21     // side of assignment)
22     *ptr = 20; // Value at address is now 20
23
24     // This prints 20
25     printf("After doing *ptr = 20, *ptr is %d\n", *ptr);
26
27     return 0;
28 }
29
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents\personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp -o basic

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic.exe
000000000061FE4C
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic1.cpp -o basic1

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic1.exe
000000000061FE44
C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C++ source file | length : 716 lines : 29 | Ln : 16 Col : 31 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic2.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

```
1  // C program to demonstrate use of * for pointers in C
2  #include <stdio.h>
3
4  int main()
5  {
6      // A normal integer variable
7      int Var = 10;
8
9      // A pointer variable that holds address of var.
10     int *ptr = &Var;
11
12     // This line prints value at address stored in ptr.
13     // Value stored is value of variable "var"
14     printf("Value of Var = %d\n", *ptr);
15
16     // The output of this line may be different in diffe
17     // runs even on same machine.
18     printf("Address of Var = %p\n", ptr);
19
20     // We can also use ptr as lvalue (Left hand
21     // side of assignment)
22     *ptr = 20; // Value at address is now 20
23
24     // This prints 20
25     printf("After doing *ptr = 20, *ptr is %d\n", *ptr);
26
27     return 0;
28 }
29
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents\personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp -o basic

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic.exe
000000000061FE4C
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic1.cpp -o basic1

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic1.exe
000000000061FE44
C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C++ source file | length : 716 lines : 29 | Ln : 16 Col : 31 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:03 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic3.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp basic1.cpp basic2.cpp basic3.cpp

```
// C++ program to illustrate Pointer Arithmetic
// in C/C++
#include <bits/stdc++.h>

// Driver program
int main()
{
    // Declare an array
    int v[3] = {10, 100, 200};

    // Declare pointer variable
    // This line prints value at address stored in ptr.
    // Value stored is value of variable "var"
    printf("Value of Var = %d\n", *ptr);

    // The output of this line may be different in different
    // runs even on same machine.
    printf("Address of Var = %p\n", ptr);

    // We can also use ptr as lvalue (Left hand
    // side of assignment)
    *ptr = 20; // Value at address is now 20

    // This prints 20
    printf("After doing *ptr = 20, *ptr is %d\n", *ptr);

    return 0;
}
```

C++ source file | length : 461 lines : 26 | Ln : 10 Col : 1 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:04 PM
3/6/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic3.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

basic.cpp | basic1.cpp | basic2.cpp | basic3.cpp

```cpp
// C++ program to illustrate Pointer Arithmetic
// in C/C++
#include <bits/stdc++.h>

// Driver program
int main()
{
    // Declare an array
    int v[3] = {10, 100, 200};

    // Declare pointer variable
    int *ptr;

    // Assign the address of v[0] to ptr
    ptr = v;

    for (int i = 0; i < 3; i++)
    {
        printf("Value of *ptr = %d\n", *ptr);
        printf("Value of ptr = %p\n\n", ptr);

        // Increment pointer ptr by 1
        ptr++;
    }
}
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.967]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents\personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp -o basic

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic.exe
000000000061FE4C
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic1.cpp -o basic1

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic1.exe
000000000061FE44
C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic2.cpp -o basic3

C:\Users\vgulia\Documents\personal_documents\C_Pointers>basic3.exe
Value of Var = 10
Address of Var = 000000000061FE44
After doing *ptr = 20, *ptr is 20

C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C++ source file | length : 461 lines : 26 | Ln : 16 Col : 1 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

7:04 PM
3/6/2019

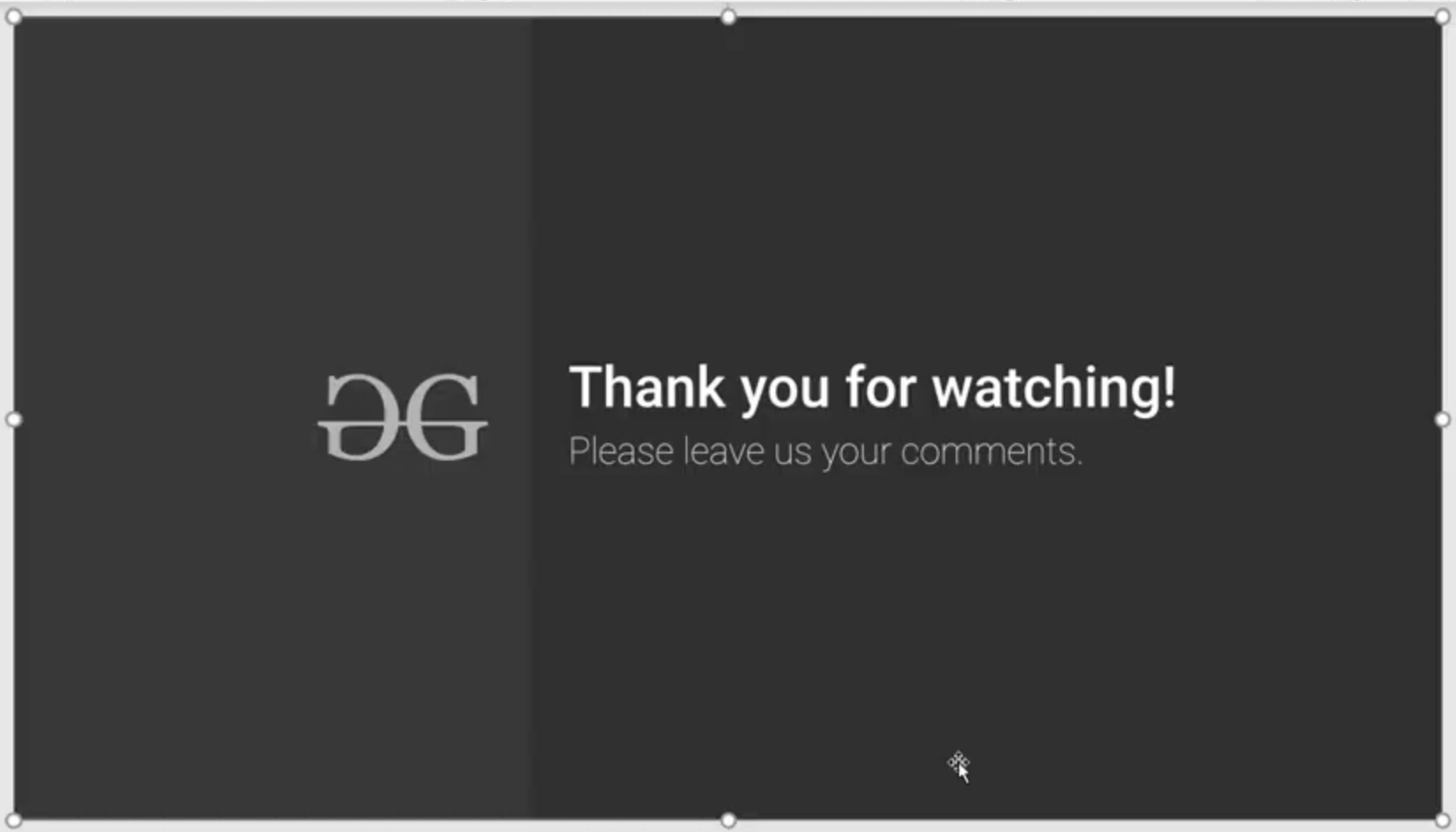
AutoSave Off | New Microsoft PowerPoint Presentation - Saved to this PC | Picture Tools | Vishal Gulia

File Home Insert Design Transitions Animations Slide Show Review View Help Storyboarding Format | Tell me what you want to do | Share | Comments

Paste | Cut | Copy | Format Painter | Clipboard
New Slide | Layout | Reset | Section | Slides
Font | Paragraph | Text Direction | Align Text | Convert to SmartArt
Arrange | Quick Styles | Shape Fill | Shape Outline | Shape Effects | Drawing
Find | Replace | Select | Editing
Dictate | Voice





Click to add notes

Slide 8 of 8 | Notes | 76%

7:06 PM
3/6/2019