

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      //variable declarations
6      int num = 5;
7      int* const ptr = &num; // Read this line from right to left => "ptr is a
8                               constant (const) pointer (*) to integer (int)."
```

9

```
10     //code
11     printf("\n");
12     printf("Current Value Of 'num' = %d\n", num);
13     printf("Current 'ptr' (Address of 'num') = %p\n", ptr);
14
15     // The following line does NOT give error ... as we are modifying the value
16     of the variable individually
17     num++;
18     printf("\n\n");
19     printf("After num++, value of 'num' = %d\n", num);
20
21     // The following line gives error and is hence commented out.
22     // We cannot alter the 'ptr' value as 'ptr' is "a constant pointer to
23     integer".
24     // With respect to the pointer, the value it points to is not constant but
25     the pointer itself is constant.
26     // Uncomment it and see the error.
27
28     // ptr++;
29
30     // The following line does NOT give error
31     // We do not get error because we are changing the value at a constant
32     pointer (address).
33     // The pointer is constant. The value to which the pointer points is NOT
34     constant.
35     (*ptr)++;
36
37     printf("\n\n");
38     printf("After (*ptr)++, value of 'ptr' = %p\n", ptr);
39     printf("Value at this 'ptr' = %d\n", *ptr);
40     printf("\n");
41     return(0);
42 }
```

39

```
40 // CONCLUSION :
41 // As "ptr" is a "constant pointer to a variable integer" - we can change the
42 value stored at address "ptr" but we cannot change the 'ptr' (Address)
43 itself.
44 // We can change the value of the variable (num) individually - whose address
45 is contained in "ptr".
46 // We can also change the "the value at address of ptr" - we can change the
47 value of "num" with respect to "ptr" => (*ptr)++ is allowed
48 // We cannot change the value OF 'ptr' => That is we cannot store a new address
49 inside 'ptr' => So, ptr++ is NOT allowed
```