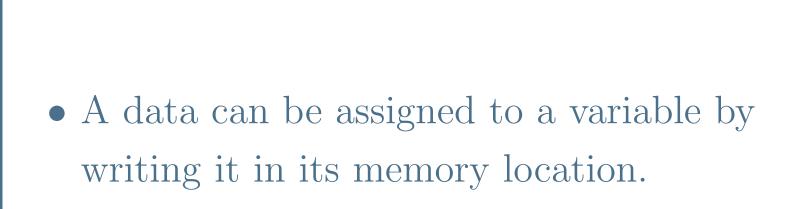


 $\mathbf{2}$

Name and Location

- The data can be read from a memory location and a data can be written in a memory location.
- These two operations are available in a high-level language.
- A variable is used as an expression, whose value is the content of the corresponding memory location.



Name and Location

- A data can be assigned to a variable by writing it in its memory location.
- But in programming languages like Python variable names are bound to objects.

Types of Data

- There arae many different types data e.g. whole numbers, integers, rational numbers, real numbers, complex numbers, vectors, matrices, characters etc.
- In the machine hardware everything is encoded as strings of binary digits (0 and 1) of finite lengths.

Types of Data

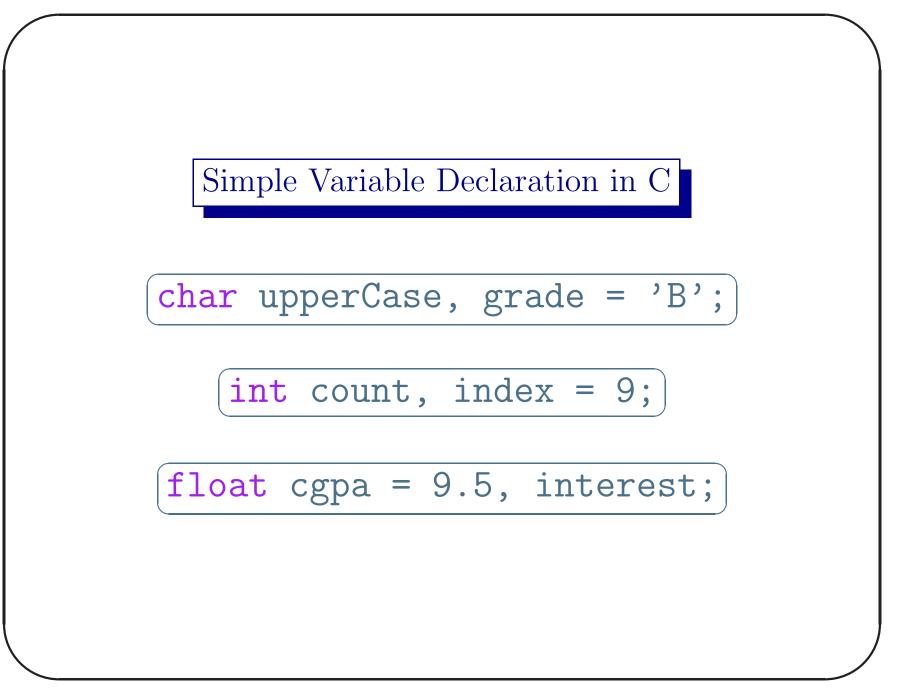
Every CPU can operate on a small set of primitive data types. The CPU instruction-set and its hardware can process these data. Finite subset of integers are processed in the ALU. Finite subset of fractional data is processes in the FPU.

Types of Data

- This often gets reflected in the built-in or primitive data types of a high level language.
- But modern high level languages go beyond that and provide built-in datatypes not directly supported by the CPU. It also provides facilities to construct new data types.



int	float	char	
unsigned int		unsigned char	
long long int	double		



- char, int, float are a few built-in data types of C language.
- upperCase and grade are variables^a of type char.
- grade = 'B' initializes the variable grade to the character code of 'B'.

^aA variable names or any C identifier follows a convention; letter, underscore followed by letter, underscore or digit.

- count and index are variables of type int.
 index = 9 initializes the variable to the binary representation of 9.
- cgpa and interest are variables of type float.

cgpa = 9.5 initializes the variable to the binary representation of 9.5 (different from int).



- An integer data may be arbitrarily large, but the C language data type int has only 32 binary digits or bit positions, for its value.
- The range of int data is
 - $-2^{31} = -2147483648$ to
 - $2^{31} 1 = 2147483647.$
- The representation is in 2's complement form.

float is an Approximation of Real

- A real numbers may have infinite information content (irrational numbers) that cannot be stored in a finite computer.
- Data type float is an approximation of real numbers. It also has a fixed 32-bit size, but the representation is different from int (IEEE 754 single precession)^a.

^aThe representations of 10 and 10.0 are different inside a computer.

Range of **float**

- The smallest and the largest magnitudes of float data are approximately
 1.401298 × 10⁻⁴⁵ and 3.402823 × 10³⁸ respectively.
- Special float values such as nan (not a number e.g. √-1) and inf (infinity 1.0/0.0) are defined to handle error in floating point operation.

char is a Short Integer

- In the binary world of computer every data, primitive or constructed, is encoded as a bit string of finite length.
- The useful set of characters are encoded as 8-bit (one byte) or 16-bit integers.
- The C language uses 8-bit ASCII^a encoding.

^aASCII stand for American Standard Code for Information Interchange.

A few ASCII Codes

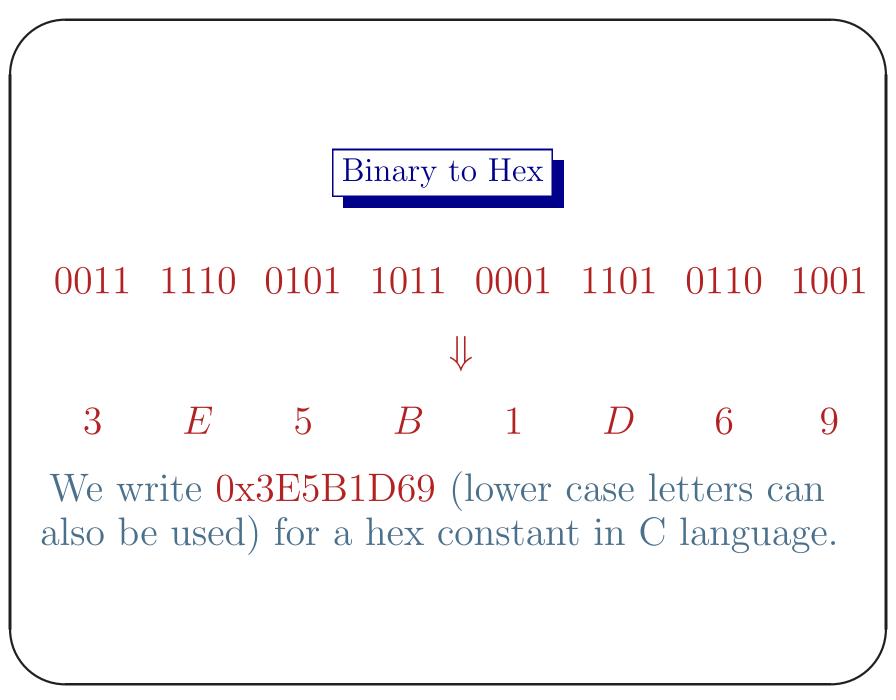
char	decimal	binary	hex
0	48	0011 0000	30
9	57	0011 1001	39
А	65	0100 0001	41
Ζ	90	0101 1010	5a
a	97	0110 0001	61
\mathbf{Z}	122	0111 1010	7a

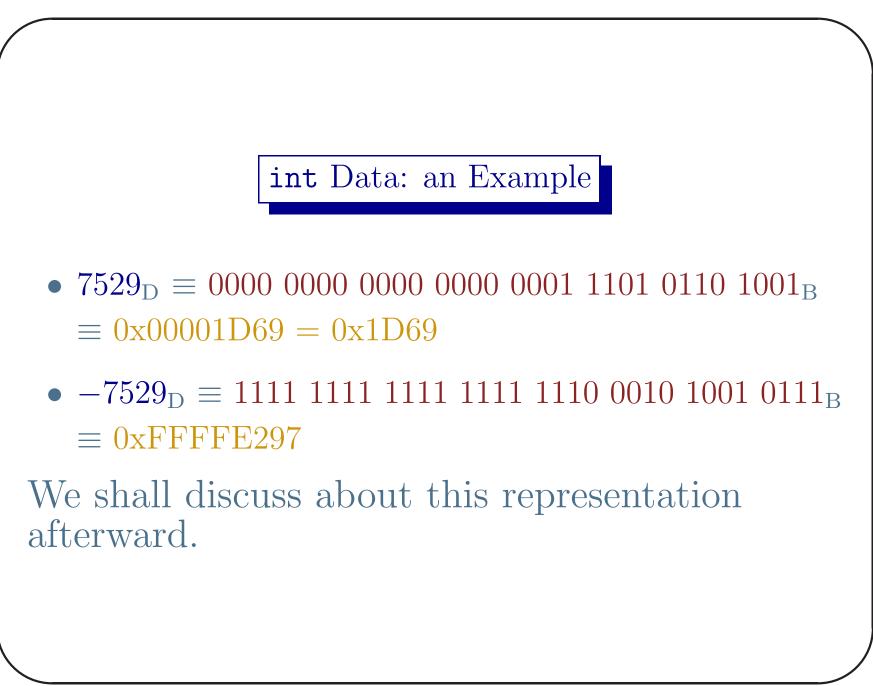


It is tedious to write a long string of binary digits. A better way is to use radix-16 or hexadecimal (Hex) number system with 16 digits $\{0, 1, \dots, 9, A(10), B(11), C(12), D(13), E(14), F(15)\}.$



To convert from binary to hex representation, the bit string is grouped in blocks of 4-bits (nibble) from the least significant side. Each block is replaced by the corresponding hex digit.

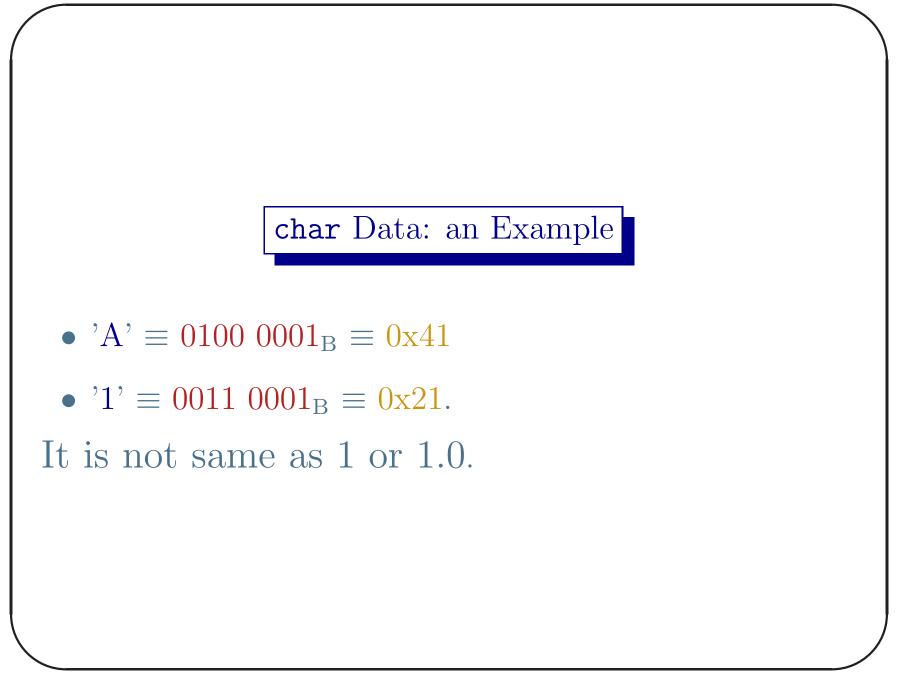




float Data: an Example

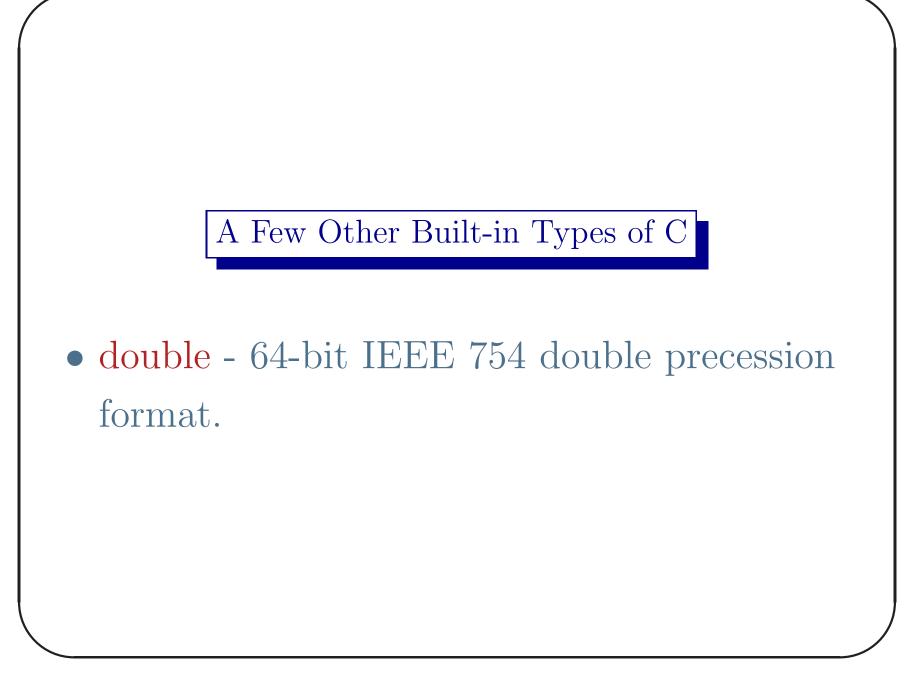
- $7529.0_{\rm D} \Rightarrow 0\ 1000\ 1011\ 110\ 1011\ 0100\ 1000\ 0000$ $0000_{\rm B}$
- $-7529.0_{\rm D} \Rightarrow 1\ 1000\ 1011\ 110\ 1011\ 0100\ 1000\ 0000$ $0000_{\rm B}$

This representations are different from that of 7529 or -7529.





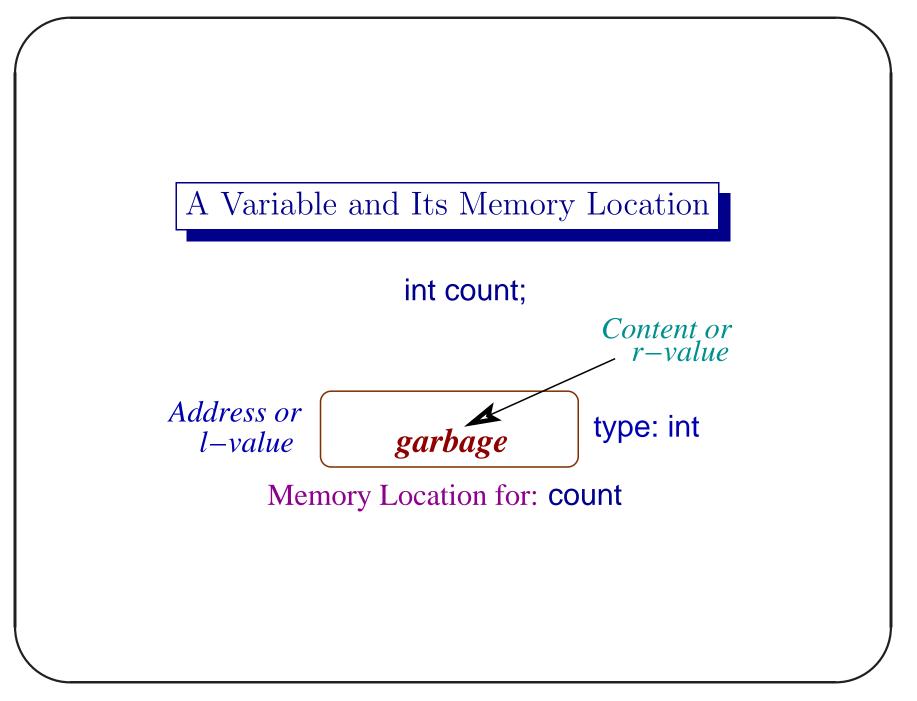
- unsigned int (unsigned) 32-bit unsigned binary, 0 to $2^{32} - 1 = 4294967295$.
- long int is same as int.
- long long int 64-bit signed binary, $-2^{63} = 9223372036854775808$ to $2^{63} - 1 = 9223372036854775807.$



Constants of Primitive Types

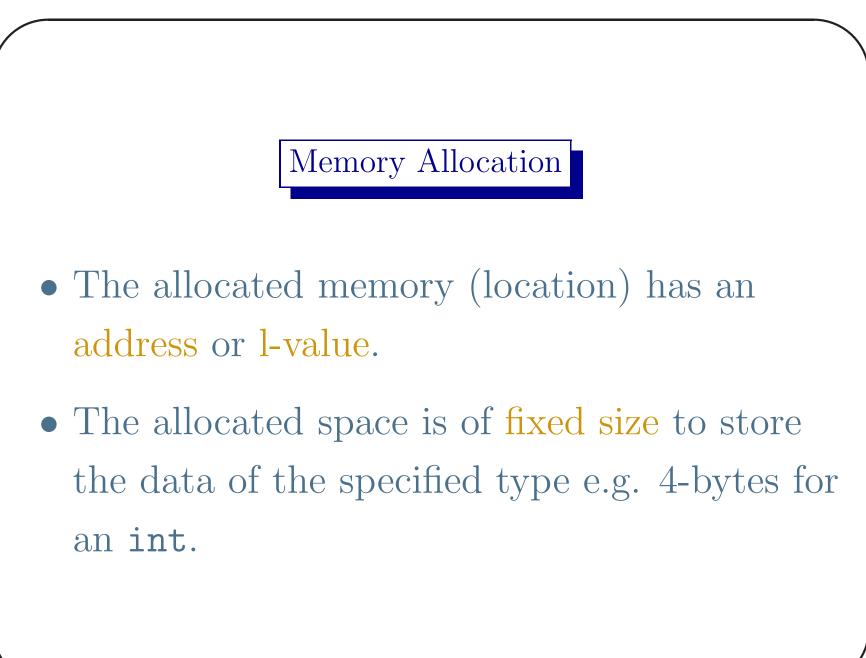
- int: 123, -123
- float: 1.23, -1.23e-02
- char: 'A', '5', '%'

A floating-point constant is often taken in double precision format.





- The compiler may generates code to allocate memory for a variable.
- For certain kind of variable the memory is allocated when the process image (a.out for example) is loaded for execution.





- Unless initialized, the content or the r-value is undefined after the declaration.
- The content or the r-value can be initialised and updated.

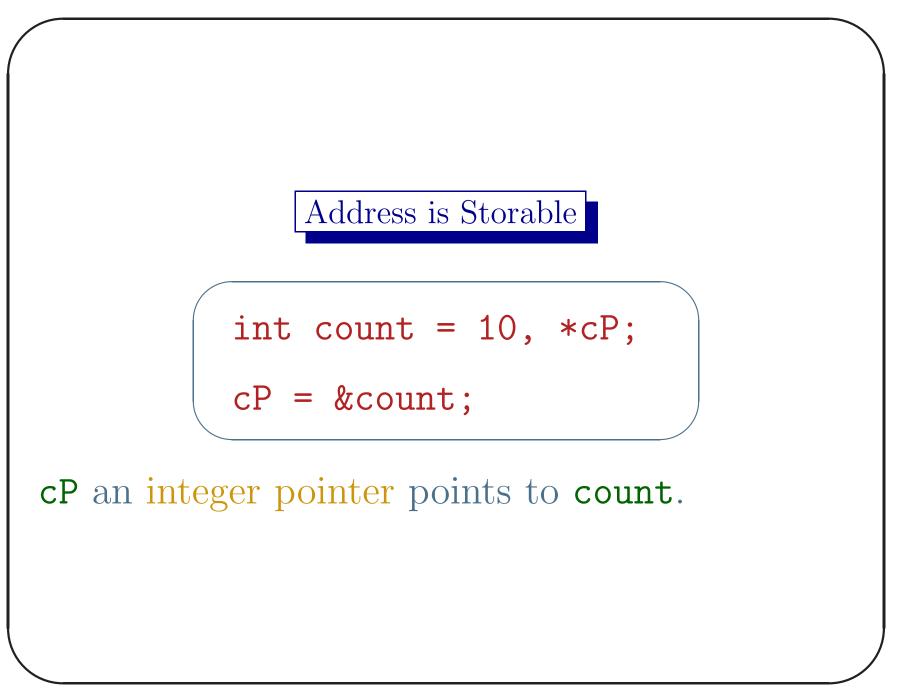
```
int count = 10;
```

```
count = 100;
```

```
count = 2*count + 5;
```

Address is Storable

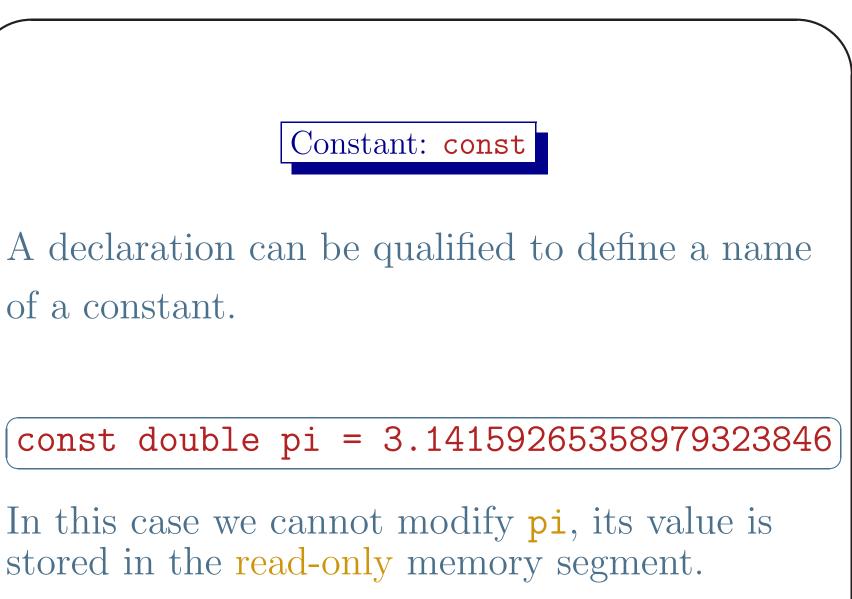
- The address or the l-value of a variable can be extracted using the unary operator '&' (&count).
- Address of a location can be stored in another variable e.g. address of an int location can be stored in a variable of type int *.



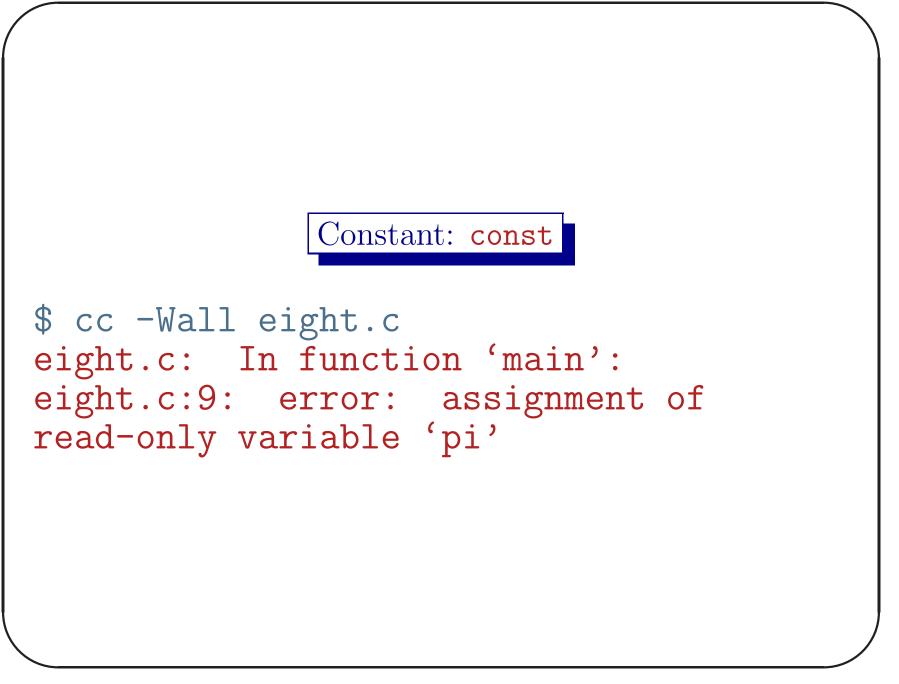
Memory Locations for Other Types

float cgpa;
char grade;

- Memory allocations are similar for other data types e.g. float and char.
- The only difference is the size (type) of the location.



```
Constant: const
#include <stdio.h>
int main()
ſ
    const double pi = 3.1415926535897932;
    pi = pi + 1;
    return 0;
} // eight.c
```



```
Reading char Data
#include <stdio.h>
int main() {
    char c, d;
    printf("Enter two characters: ");
    scanf("%c", &c);
    scanf("%c", &d);
    printf("c.., cn", c, d);
    return 0;
} // charRead.c
This program is expected to read two
characters from two lines.
```

```
Reading char Data
$ cc -Wall charRead.c
$ a.out
Enter two characters: 1
1..
$
It does not read the second character. The
reason is that pressing of 'Enter' key injects a
non-printable character '\n' in the input
stream.
```

