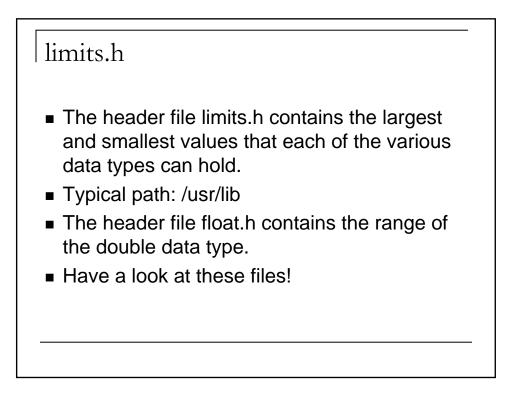


Continuing with Data types...

Input Output for short and long int #include<stdio.h> void main() { short int shorti; long int longi; printf("Input short int: "); scanf("%hd",&shorti); printf("%hd\n",shorti); printf("Inputt long int: "); scanf("%ld",&longi); printf("%ld\n",longi); printf("shorti = %hd and longi=%ld",shorti,longi); } A Sample Run: Input short int: 20 Input long int: 2000000 shorti= 20 and longi= 2000000

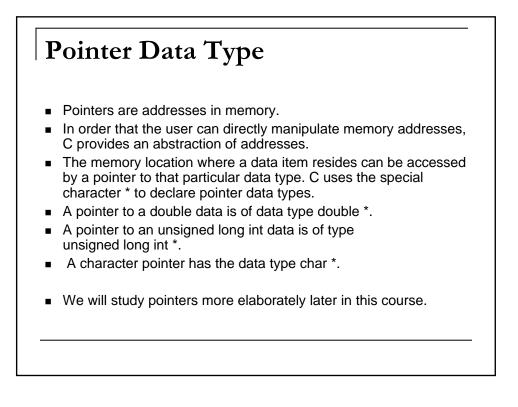


The typedef statement

- This statement can be used to define new data types.
 - □ For example:
 - typedef unsigned long ulong;
 - ulong is a new data type equivalent to unsigned long
 - It can be used as any other data type as follows; ulong u;

(declares u to be of the type ulong)

 The size of the new data type can also be found in bytes using sizeof(ulong)

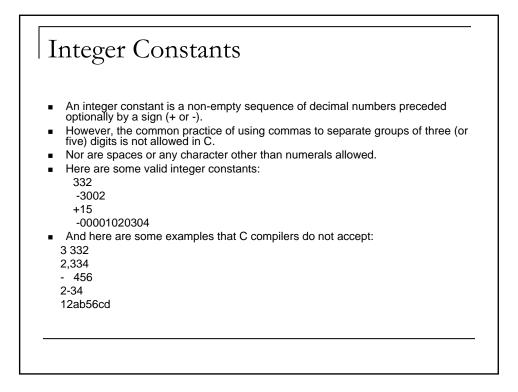


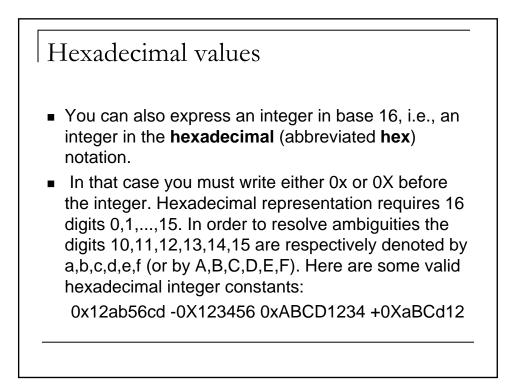
Examples

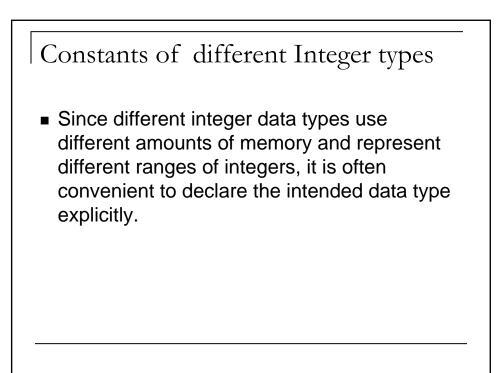
- int m, n, armadillo;
- int platypus;
- float hi, goodMorning;
- unsigned char _u_the_charcoal;

Constants

- Defining a data type is not enough.
- You need to assign the variables and work with specific values of various data types.
- Examples: PI (hopefully it will not change its value!)
- 1.0/n is our previous example of finding reciprocals has a constant.



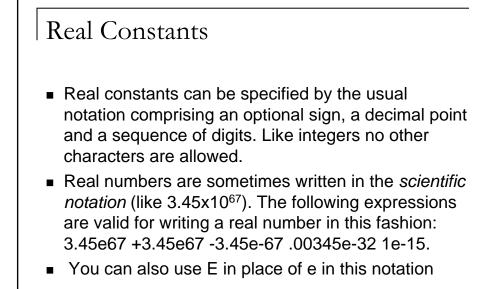


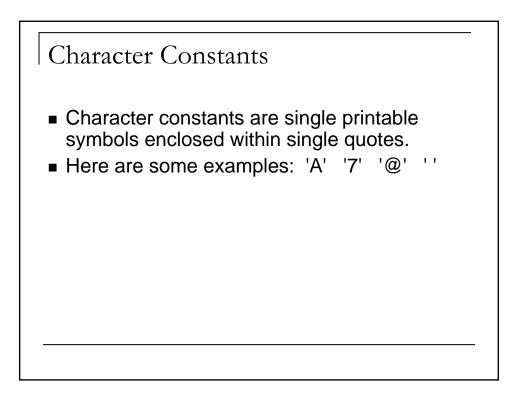


Suffix	Data type	
L (or I)	long	
LL (or II)	long long	
U (or u)	unsigned	
UL (or ul)	unsigned long	
ULL (or ull)	unsigned long long	

Examples

- 400000000UL
- 123U
- 0x7FFFFFFF
- 0x123456789abcdef0ULL

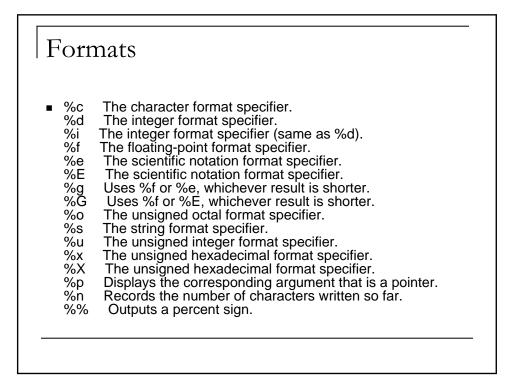


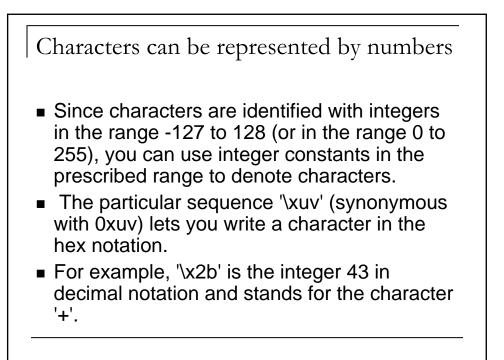


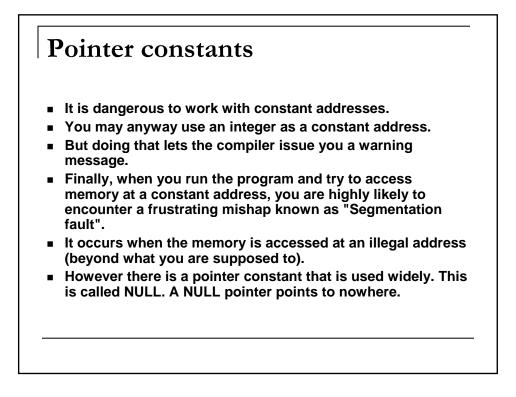
Special Characters			
Constant	Character	ASCII value	
'\0'	Null	0	
'\b'	Backspace	8	
'\t'	Tab	9	
'\n'	New line	13	
'\''	Quote	39	
'\\'	Backslash	92	

```
Try this!
```

```
#include<stdio.h>
main()
{
    int i;
    for(i=0;i<10000;i++)
    prinf("%c",'\a')
}</pre>
```

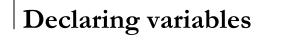




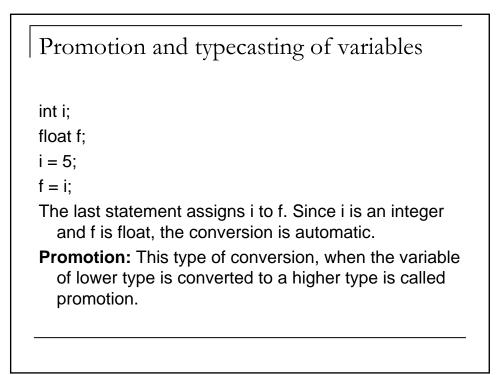


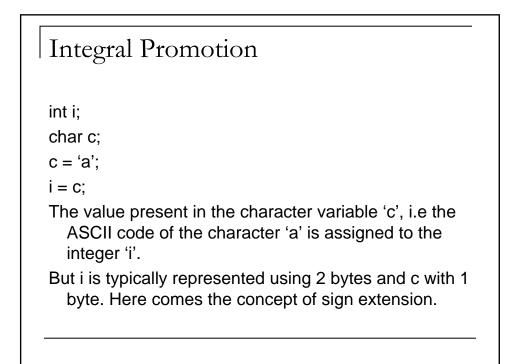
Variables

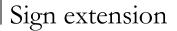
- "The only constant thing is change"
- Variables help to abstract this change.
- Teacher = XYZ ;
 - here Teacher is a variable
 - XYZ is one instance of the variable, and is a constant
- A variable is an entity that has a value and is known to the program by a name,
- A variable definition associates a memory location with the variable name.
- At one time it can have only one value associated with it.



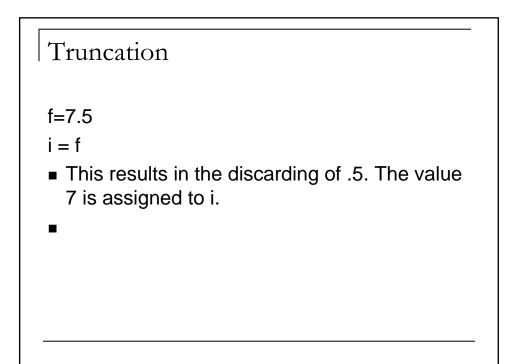
- For declaring one or more variables of a given data type do the following:
- First write the data type of the variable.
- Then put a space (or any other white character).
- Then write your comma-separated list of variable names.
- At the end put a semi-colon.







- Conversion to a signed integer from character data type:
 - □ lower 8 bits will be the character's value.
 - higher 8 bits will be filled with 0 or 1, depending on the Maximum Significant Bit (MSB) of the character.
 - (Note: MSB is used to indicate the sign of a signed number)
- This is called sign extension.
- Sign extension takes place only if the variable of the higher type is signed.



Forcible Conversion

