Programming for Engineers

Data Types

UNIVERSITY AT ALBANY
State University of New York

ICEN 200 – Spring 2018

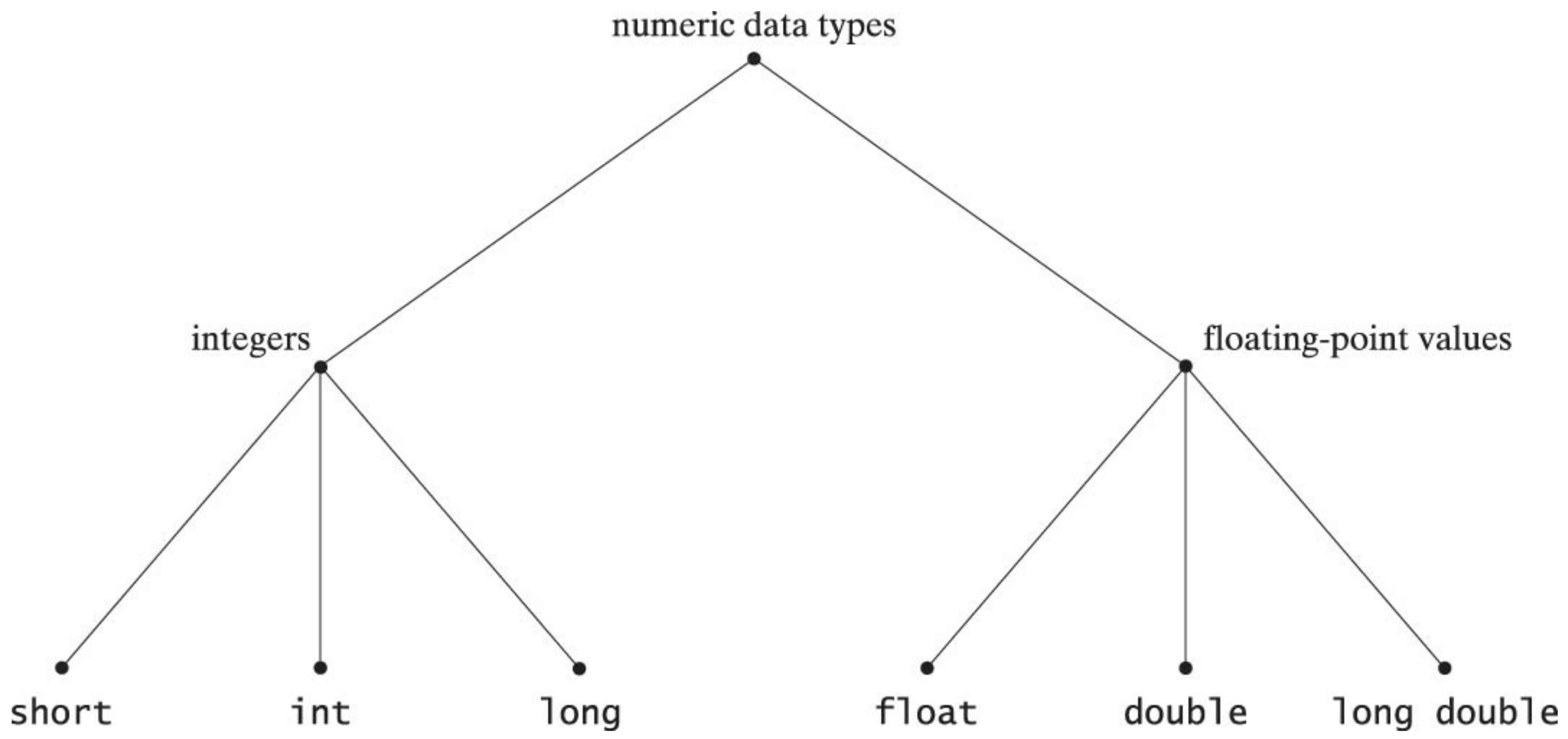
Prof. Dola Saha

UNIVERSITY AT ALBANY
State University of New York

Data Types

Data Type	Description	Bytes in Memory
`char`	Character	1
`int`	Whole number	4 or 2 (natural size of integer in host machine)
`float`	Real number - Single precision floating point	Usually 4
`double`	Real number - Double precision floating point	Usually 8
`short`	Shorter than regular	Usually 2
`long`	Longer than regular	Usually 8
`unsigned`	No bits used for sign	
`signed`	1 bit used for sign	

Numeric Data Types



Data type: char

- 1 Byte or 8 bits
- Example: A, c, x, q
- Character is represented in memory as a binary number
- Value stored is determined by ASCII (American Standard Code for Information Interchange) code.
- Print format: %c
- If printed with %d
  - Prints the value in ASCII

Character	ASCII Code
`' '`	32
`'*'`	42
`'A'`	65
`'B'`	66
`'Z'`	90
`'a'`	97
`'b'`	98
`'z'`	122
`'0'`	48
`'9'`	57

Character and ASCII

```c
#include <stdio.h>

//function main begins program execution
int main()
{
    char myChar; //character variable

    // Get a character from user
    printf("Enter a character: ");
    scanf("%c", &myChar);

    // Print the character in ASCII
    printf("The ASCII form of %c is %d\n", myChar, myChar);

}
// end function main()
```

```
Enter a character: A
The ASCII form of A is 65
```

Data type: `int`

- Standard Integer
- Limited by size of memory
- Usually 4 bytes
- Value stored in binary
- 1 bit for sign (0 for positive, 1 for negative)
- Range: -2147483648, 2147483647
- Print format: %d
- Use unsigned to use all the bits

Integer will not suffice – real applications

42.686908, -73.823919
University Art Museum, University at Albany
UAlbany Performing Arts Center
University at Albany
State Quad
Dutch Quad
Indian Quad
SEFCU Arena
Lacrosse Field (John Fallon Field)
Field Hockey - Alumni Turf
Campus Center
Albany, NY 12206
42.686908, -73.823919

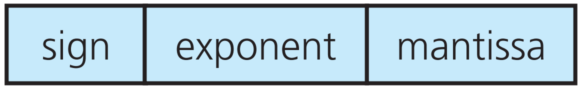
- Calculate area of a circle
- Calculate average of grades in class

UNIVERSITY AT ALBANY
State University of New York

Float, Double

- Real number, analogous to scientific notation
- Storage area divided into three areas:
  - Sign (0 for positive, 1 for negative)
  - Exponent (repeated multiplication)
  - Mantissa (binary fraction between 0.5 and 1)

type `double` format



- The mantissa and exponent are chosen such that the following formula is correct

$$real\ number = mantissa \times 2^{exponent}$$

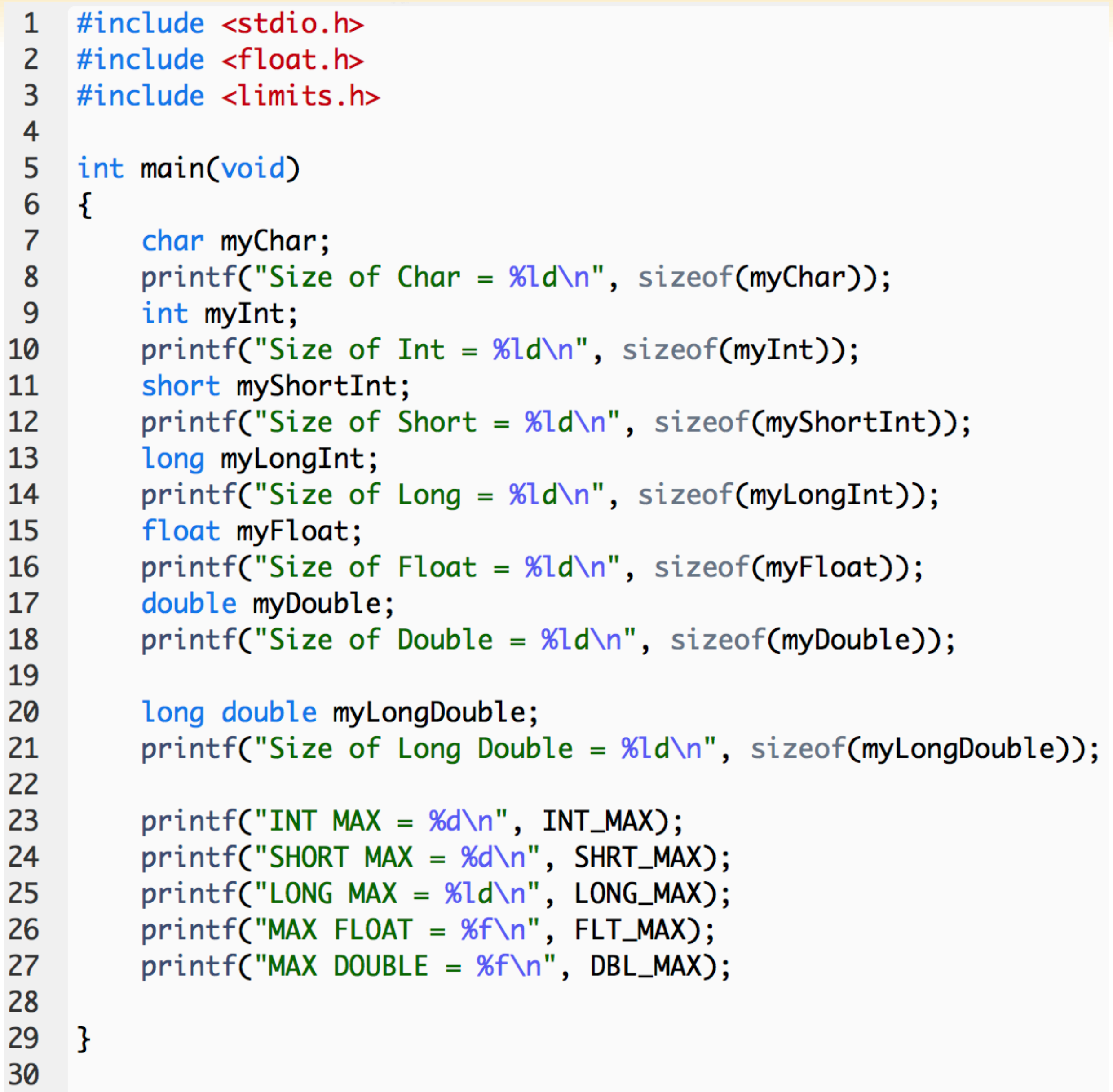
Float, Double

- Float (single precision)
  - 1 bit sign, 8 bits exponent, 23 bits mantissa
- Double (double precision)
  - 1 bit sign, 11 bits exponent, 52 bits mantissa
- Depends on hardware
- Print format: %f (for float) %lf (for double)

Short, Long, Long Double

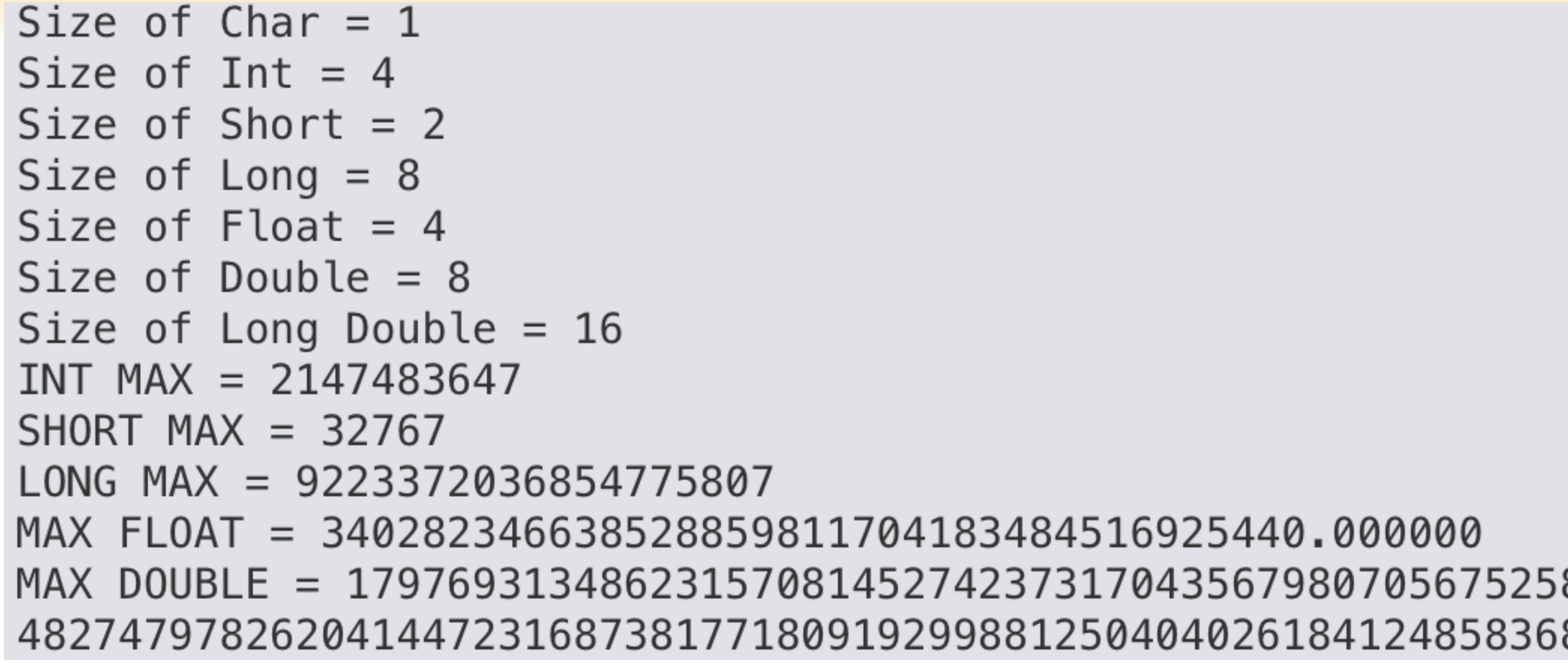
- Short
  - Usually 2 bytes whole number
  - Print format: %d
- Long
  - Usually 8 bytes whole number
  - Print format: %ld
- Long Double
  - Usually 16 bytes fractional
  - Print format: %Lf

Size and limits



```c
#include <stdio.h>
#include <float.h>
#include <limits.h>

int main(void)
{
    char myChar;
    printf("Size of Char = %ld\n", sizeof(myChar));
    int myInt;
    printf("Size of Int = %ld\n", sizeof(myInt));
    short myShortInt;
    printf("Size of Short = %ld\n", sizeof(myShortInt));
    long myLongInt;
    printf("Size of Long = %ld\n", sizeof(myLongInt));
    float myFloat;
    printf("Size of Float = %ld\n", sizeof(myFloat));
    double myDouble;
    printf("Size of Double = %ld\n", sizeof(myDouble));

    long double myLongDouble;
    printf("Size of Long Double = %ld\n", sizeof(myLongDouble));

    printf("INT MAX = %d\n", INT_MAX);
    printf("SHORT MAX = %d\n", SHRT_MAX);
    printf("LONG MAX = %ld\n", LONG_MAX);
    printf("MAX FLOAT = %f\n", FLT_MAX);
    printf("MAX DOUBLE = %f\n", DBL_MAX);

}
```

Output of size



```
Size of Char = 1
Size of Int = 4
Size of Short = 2
Size of Long = 8
Size of Float = 4
Size of Double = 8
Size of Long Double = 16
INT MAX = 2147483647
SHORT MAX = 32767
LONG MAX = 9223372036854775807
MAX FLOAT = 340282346638528859811704183484516925440.000000
MAX DOUBLE = 17976931348623157081452742373170435679807056752584
4827479782620414472316873817718091929988125040402618412485836
```

Ranges

Whole Number

Type	Range in Typical Microprocessor Implementation
`short`	−32,767 .. 32,767
`unsigned short`	0 .. 65,535
`int`	−2,147,483,647 .. 2,147,483,647
`unsigned`	0 .. 4,294,967,295
`long`	−2,147,483,647 .. 2,147,483,647
`unsigned long`	0 .. 4,294,967,295

Real Number

Type	Approximate Range*	Significant Digits*
`float`	10^{-37} .. 10^{38}	6
`double`	10^{-307} .. 10^{308}	15
`long double`	10^{-4931} .. 10^{4932}	19

*In a typical microprocessor-based C implementation

Review Questions

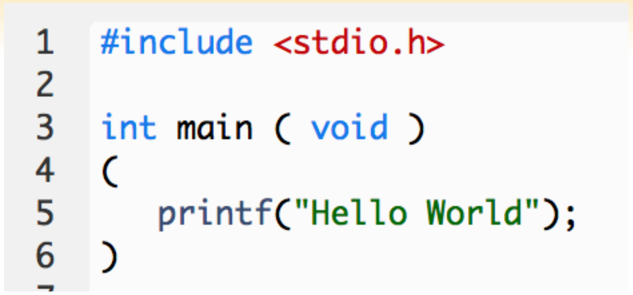
- State True or False:
  - Short takes more memory space than Integer (int)
  - Float and double are real number representations in C
  - Char is represented in memory by ASCII
  - Print format for char is %d
  - Print format for double is %lf
  - Float and double has 2 parts: exponent and mantissa

Review Questions / Answers

- State True or False:

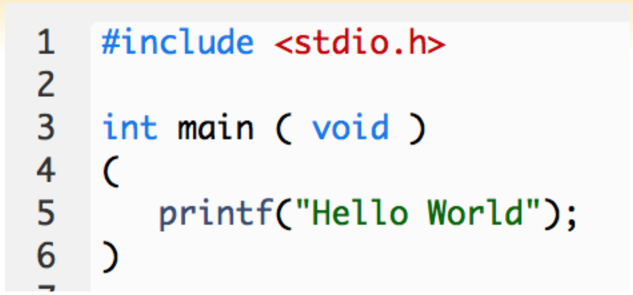
Short takes more memory space than Integer (int)	FALSE
Float and double are real number representations in C	TRUE
Char is represented in memory by ASCII	TRUE
Print format for char is %d	FALSE
Print format for double is %lf	TRUE
Float and double has 2 parts: exponent and mantissa	FALSE

What is the error in code?



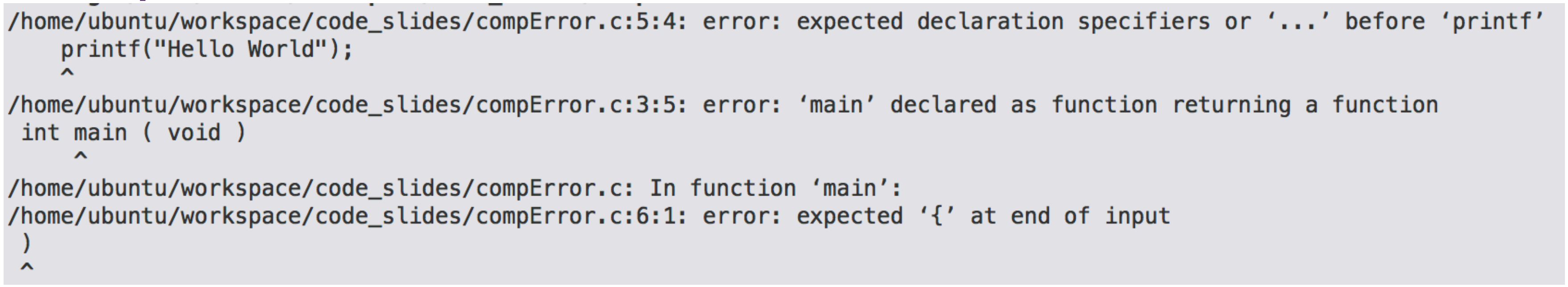
```c
#include <stdio.h>

int main ( void )
(
    printf("Hello World");
)
```

What is the error in code?



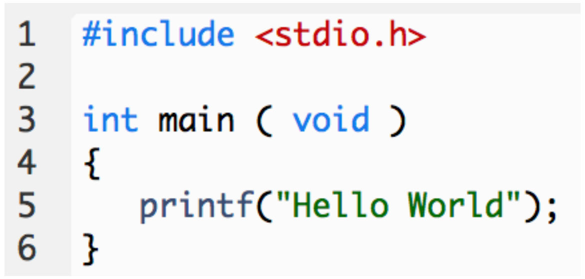
```
1  #include <stdio.h>
2
3  int main ( void )
4  (
5     printf("Hello World");
6  )
```

Compilation Error



```
/home/ubuntu/workspace/code_slides/compError.c:5:4: error: expected declaration specifiers or ‘...’ before ‘printf’
    printf("Hello World");
    ^
/home/ubuntu/workspace/code_slides/compError.c:3:5: error: ‘main’ declared as function returning a function
 int main ( void )
     ^
/home/ubuntu/workspace/code_slides/compError.c: In function ‘main’:
/home/ubuntu/workspace/code_slides/compError.c:6:1: error: expected ‘{’ at end of input
 )
 ^
```

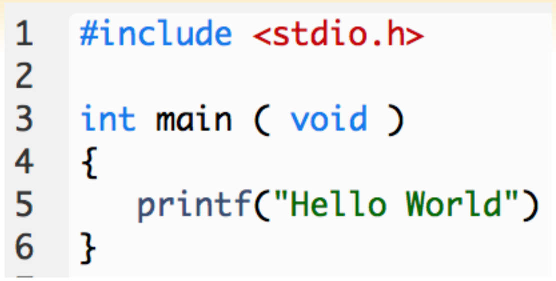
Correct Code



```
1  #include <stdio.h>
2
3  int main ( void )
4  {
5     printf("Hello World");
6  }
```

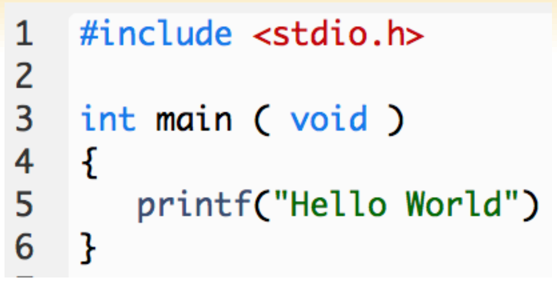
UNIVERSITY AT ALBANY
State University of New York

What is the error in code?



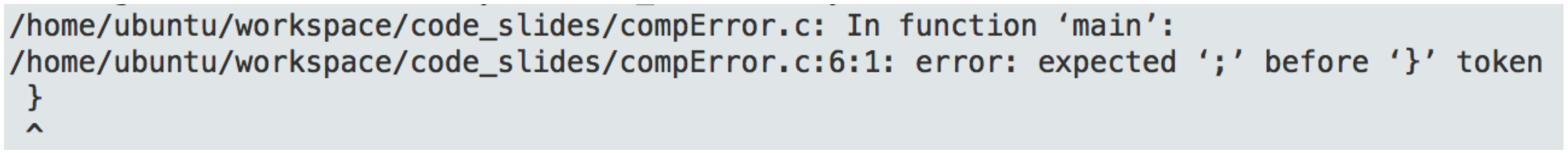
```c
#include <stdio.h>

int main ( void )
{
    printf("Hello World")
}
```

What is the error in code?



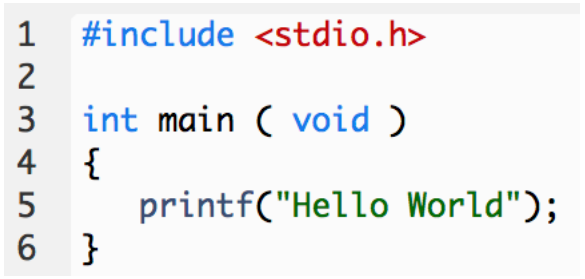
```c
1  #include <stdio.h>
2
3  int main ( void )
4  {
5      printf("Hello World")
6  }
```

Compilation Error



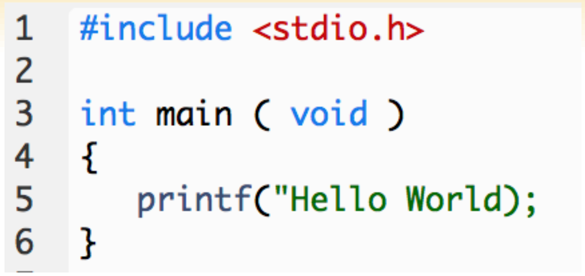
```
/home/ubuntu/workspace/code_slides/compError.c: In function ‘main’:
/home/ubuntu/workspace/code_slides/compError.c:6:1: error: expected ‘;’ before ‘}’ token
 }
 ^
```

Correct Code



```c
1  #include <stdio.h>
2
3  int main ( void )
4  {
5      printf("Hello World");
6  }
```

What is the error in code?



```
#include <stdio.h>

int main ( void )
{
    printf("Hello World);
}
```

UNIVERSITY AT ALBANY
State University of New York

What is the error in code?

```c
#include <stdio.h>

int main ( void )
{
    printf("Hello World);
}
```

Compilation Error

```
/home/ubuntu/workspace/code_slides/compError.c: In function ‘main’:
/home/ubuntu/workspace/code_slides/compError.c:5:11: warning: missing terminating " character [enabled by default]
    printf("Hello World);
           ^
/home/ubuntu/workspace/code_slides/compError.c:5:4: error: missing terminating " character
    printf("Hello World);
    ^
/home/ubuntu/workspace/code_slides/compError.c:6:1: error: expected expression before ‘}’ token
 }
 ^
/home/ubuntu/workspace/code_slides/compError.c:6:1: error: expected ‘;’ before ‘}’ token
```

Correct Code

```c
#include <stdio.h>

int main ( void )
{
    printf("Hello World");
}
```

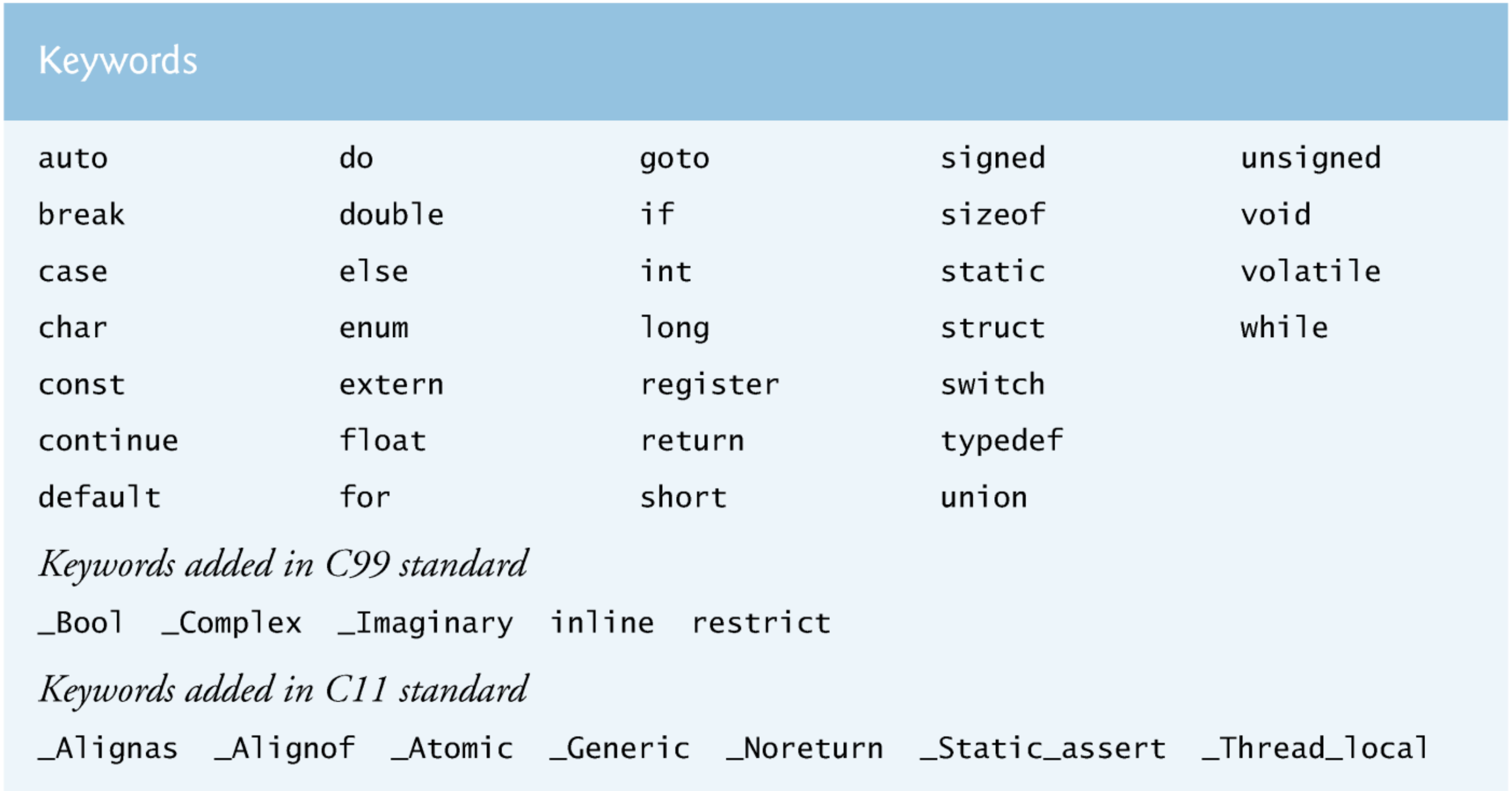
UNIVERSITY AT ALBANY
State University of New York

Common Errors

- Omitting the parentheses after main.
- Omitting or incorrectly typing the opening brace { that signifies the start of a function body.
- Omitting or incorrectly typing the closing brace } that signifies the end of a function.
- Misspelling the name of a function; for example, typing pintf ( ) instead of printf ( ).
- Forgetting to close the message to printf ( ) with a double quote symbol.
- Omitting the semicolon at the end of each C statement.
- Adding a semicolon at the end of the #include preprocessor command.
- Forgetting the \n to indicate a new line.
- Incorrectly typing the letter O for the number zero (0), or vice versa.
- *Incorrectly typing the letter l for the number 1, or vice versa.*

C Keywords

- Reserved words of the language, special meaning to C compiler
- Do not use these as identifiers, like variable names



Keywords

`auto`	`do`	`goto`	`signed`	`unsigned`
`break`	`double`	`if`	`sizeof`	`void`
`case`	`else`	`int`	`static`	`volatile`
`char`	`enum`	`long`	`struct`	`while`
`const`	`extern`	`register`	`switch`	
`continue`	`float`	`return`	`typedef`	
`default`	`for`	`short`	`union`	

Keywords added in C99 standard

`_Bool` `_Complex` `_Imaginary` `inline` `restrict`

Keywords added in C11 standard

`_Alignas` `_Alignof` `_Atomic` `_Generic` `_Noreturn` `_Static_assert` `_Thread_local`