Data Conversion

software diversity

in acction

When is Conversion Needed

- Mixed Type Expressions

  ```
  int x; float y; x=y*x;
  ```

- Assignment Statements

  ```
  int x; float y; x=y*3.0;
  ```

- Argument Evaluation

  ```
  int myfn(float x); int y=myfn(3);
  ```

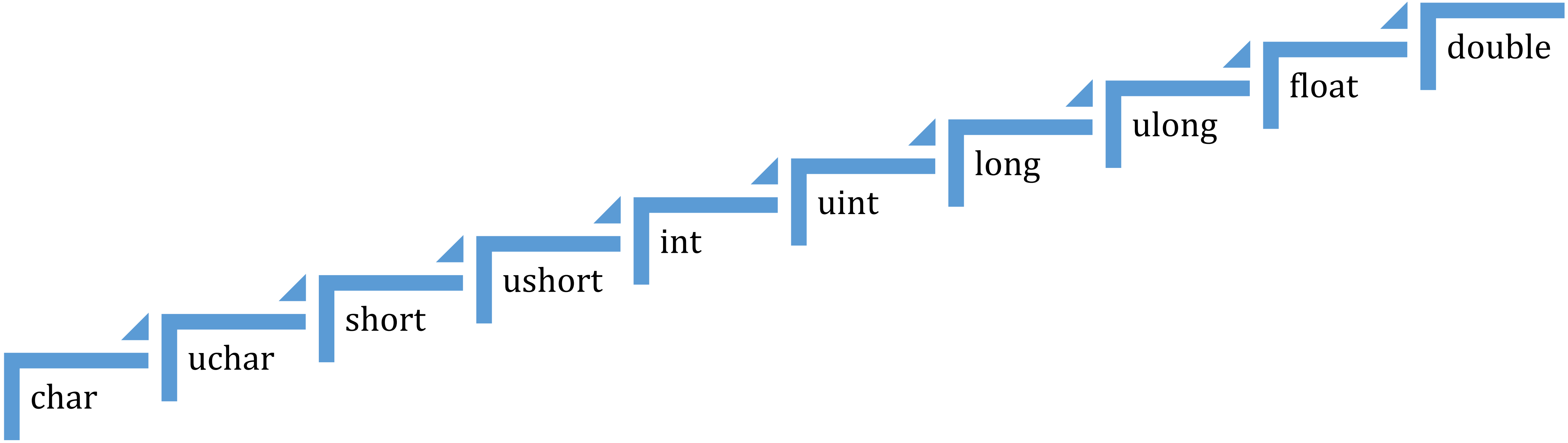
- Explicit Casting

  ```
  int x=7; float y = ((float)x)/3;
  ```

C Automatic type conversion rules

- In an expression, C converts all components in that expression to the most “general” type, and then evaluates the expression using that general type
- In an assignment (or argument evaluation), C converts the value of the expression to the type of the receiver

Generality of Numeric Types



Conversion Strategies…

- signed vs. unsigned
  - Interpret the same bits a different way… 0xFF = 256 unsigned, = -1 signed
- short integer vs. long integer
  - Signed types
    - Pad on left with sign bit 0xFF -> 0xFFFF FFFF or truncate 0x0000 007C -> 0x7C
  - Unsigned types
    - Pad on left with 0 0xFF->0x0000 00FF or truncate
- Integer vs. Float
  - Float -> Integer… drop .<xxx> (round towards 0)
  - Integer -> Float… add .0, and round to nearest floating point value

Conversion Errors

		TO TYPE									
		char	uchar	short	ushort	int	uint	long	ulong	float	double
FROM TYPE	char		🟢	🟩	🟢	🟩	🟢	🟩	🟢	🟩	🟩
	uchar	🟨		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
	short	🟨	🟪		🟢	🟩	🟢	🟩	🟢	🟩	🟩
	ushort	🟨	🟨	🟨		🟩	🟩	🟩	🟩	🟩	🟩
	int	🟨	🟪	🟨	🟪		🟢	🟩	🟢	🟧	🟩
	uint	🟨	🟨	🟨	🟨	🟨		🟩	🟩	🟧	🟩
	long	🟨	🟪	🟨	🟪	🟨	🟪		🟢	🟧	🟧
	ulong	🟨	🟨	🟨	🟨	🟨	🟨	🟨		🟧	🟧
	float	🟥	▨	🟥	▨	🟥	▨	🟥	▨		🟩
	double	🟥	▨	🟥	▨	🟥	▨	🟥	▨	🟥	

LEGEND	
🟩	No Error
🟢	Wrong if <0
🟨	Wrong if too big
🟪	Wrong if <0 or too big
🟧	Rounded +/-
🟥	Rounded +/- Wrong if too big
▨	Wrong if <0, too big, rounded +/-

Examples of Errors

🟩	No Error	char a=14; int b=a;
🟩	Wrong if <0	char a= −8; unsigned short b= a; // a=xF8, b = xFFF8 = 65,528
🟨	Wrong if too big	short a=217; char b=a; // a=x00D9, b=xD9 = −39

🟪	Wrong if <0 or too big	int a=−8; unsigned short b=a; // a=xFFFF FFF8, b=FFF8 = 65,258
		int a=393,659 unsigned short b=a; // a=x0006 01BB, b=01BB = 443

🟧	Rounded +/-	int a=100000001; float b=a; // b=1e8

🟥	Rounded +/-
	Wrong if too big
		float a=3.17; int b=a; // b=3
		float a=3e2; char b=a; // a=300.0 = 0x012C, b=x2C =44

🟪	Wrong if <0, too big, rounded +/-	float a=−317.3; unsigned char b=a; // a=xFEC3, b=xC3 = 195

Explicit Casting

- Programmer tells C explicitly to perform conversion
- "cast" prefix operator (<type>)<expression>
  - Causes expression to be evaluated and then converted to the specified type
- Needed when the programmer knows better than the compiler!

Resources

- Programming in C, Chapter 3, 13 (pp 325-328)
- WikiPedia: Operators in C and C++ (https://en.wikipedia.org/wiki/Operators_in_C_and_C%2B%2B)
- WikiPedia: C Data Types (https://en.wikipedia.org/wiki/C_data_types)
- Wikipedia: Short Circuit Evaluation (https://en.wikipedia.org/wiki/Short-circuit_evaluation)