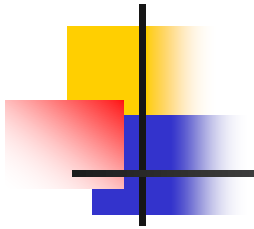
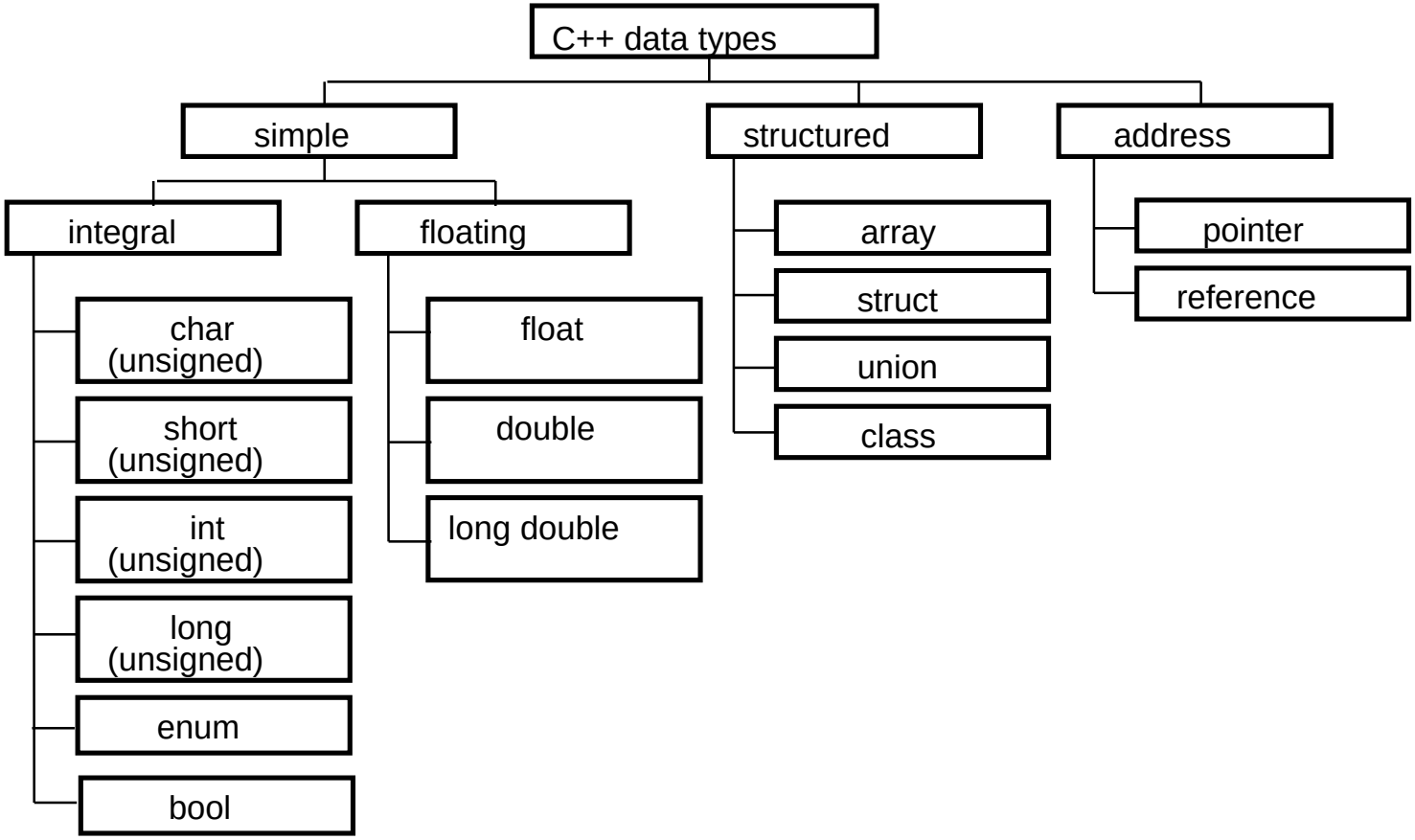
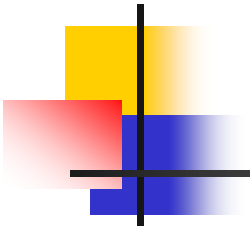


C++ Review

- C++ Programming Language
  - Data Types
  - Streams and Files
  - Process Structures

C++ Data Types


C++ data types
simple
structured
address
integral
floating
array
pointer
struct
reference
union
class
char (unsigned)
short (unsigned)
int (unsigned)
long (unsigned)
enum
bool
float
double
long double

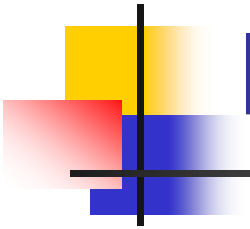


Integral Data Types

- Integral

```
1<=sizeof(char)<=sizeof(short)
   <=sizeof(int)<=sizeof(long)
```

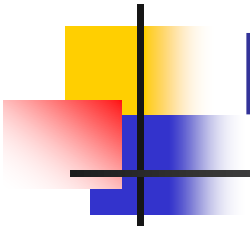
  - signed range
    - minimum value $-(2^{(numbits - 1)})$
    - maximum value $2^{(numbits - 1)} - 1$
  - unsigned range
    - minimum value 0
    - maximum value $2^{(numbits)} - 1$



Enumerated Types

- User defined type consisting of named constants called enumerators
  - by default, first enumerator has value 0 unless explicitly reset
  - can be promoted to integer value; integer must be cast as enumerated type (results undefined)

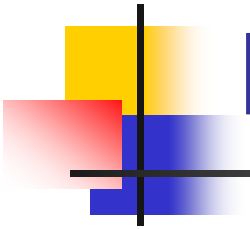
```
enum Day {Sunday = 1, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday};

// declare some local variables
Day yesterday, today, tomorrow;
yesterday = Sunday;
today = static_cast<Day>(2);
tomorrow = static_cast<Day>(today + 1);
```



Boolean Types

- Integral type having values true or false
  - all conditional expressions return value of type bool
  - allows you to create small integer variables that are suitable for holding true or false values
    - true → 1
    - false → 0

```
int inpNum;
bool isOdd;
isOdd = (inpNum % 2) != 0;
```



Floating Data Types

- Floating

`sizeof(float)<=sizeof(double)<=sizeof(long double)`

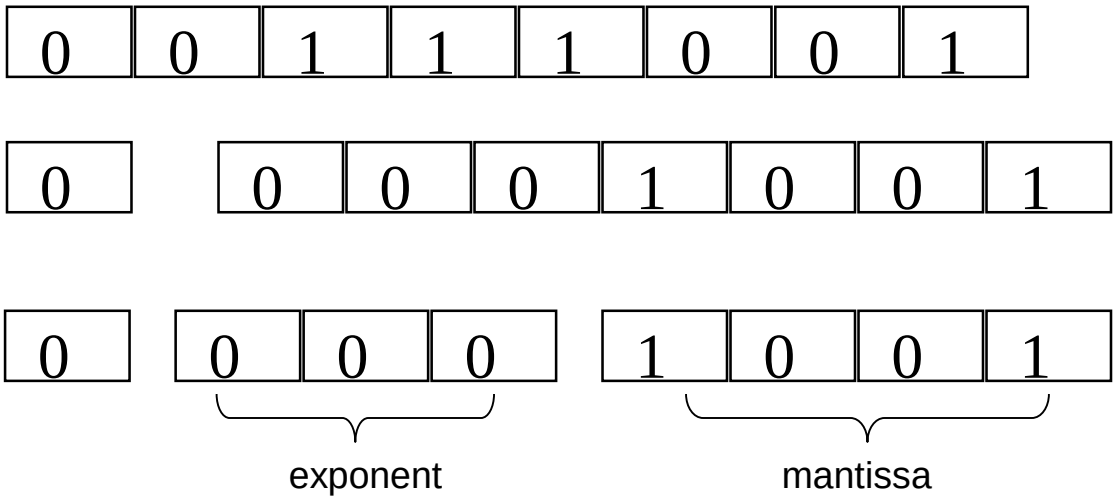
  - expressed in exponential (scientific) notation
    - exponent determines range
    - fraction or mantissa determines precision

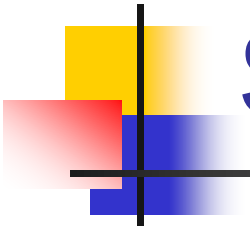
Decimal Notation	**Scientific Notation**
247.91	2.4791 x 10^2
0.00072	7.2 x 10^(-4)

Internal Data Representation

- internal binary storage depends upon data type

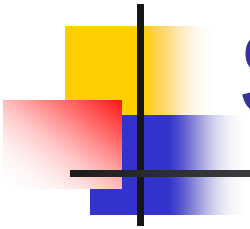
```
char cVar = '9';
   (ASCII 57)
int iVar = 9;

float fVar = 9.0;
```





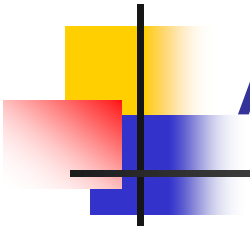
Structured Data Types

- Array
  - finite sequential list of homogeneous data
  - elements directly accessed by index
  - array name is constant pointer to first array element
  - **usually no checking is done to verify that index or pointer is in array bounds--program may overwrite other variables**
  - 2D array created by nesting 1D arrays
    - stored in row order
  - **C string** is a NULL-terminated character array



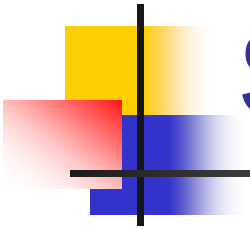
Structured Data Types

- Struct
  - record that bundles heterogeneous data
  - fields accessed by name
    - by default all members are public
- Union
  - user defined data type that can hold different types at different times
  - user must keep track of current contents and access them appropriately
- Class



Address Data Types

- Pointer
  - unsigned integer that represents memory address
  - arithmetic operators update pointer variable to reference next item of that type
  - need dereferencing operators to access underlying data
- Reference
  - like pointer, represents memory address
  - serves as alias to another variable
    - dereferencing operators not needed to access underlying data

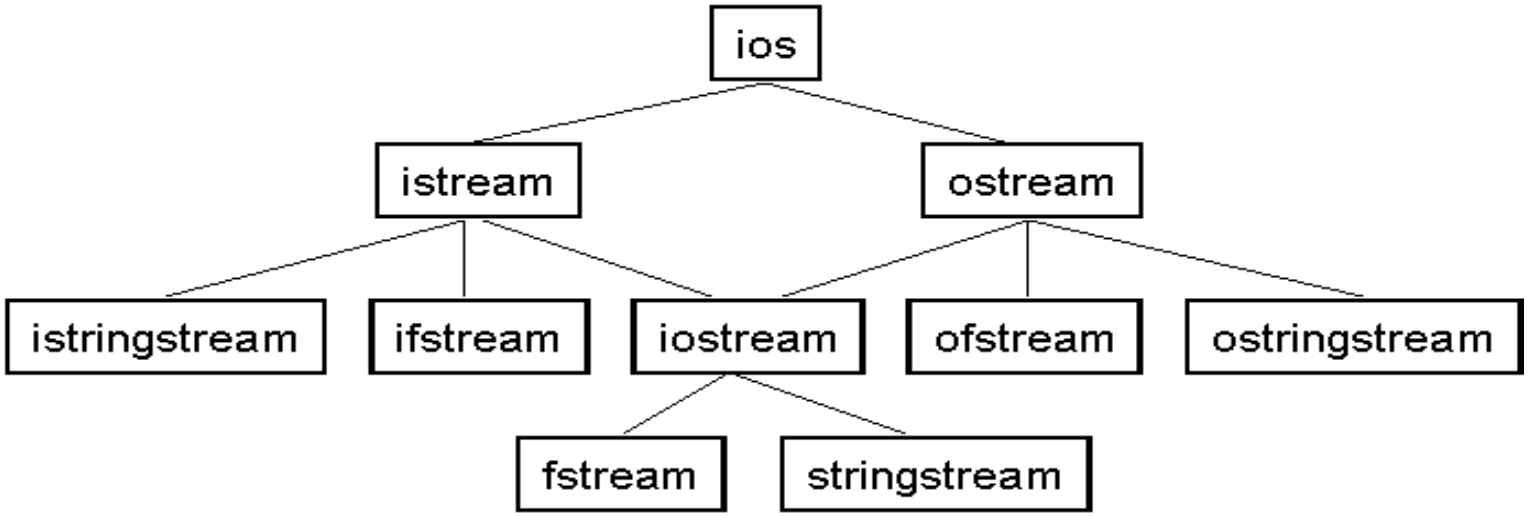


Streams and Files

- Data stream is logically connected to a file
  - read-only
  - write-only
  - read-write
- File access
  - sequential
  - random
- Stream maintains **file pointer** that identifies current position in the stream
  - disk filetypes
    - text contains printable ASCII characters
    - binary contains pure binary data

Stream Classes

- stream header files
  - #include <iostream>
    - input and output stream class
  - #include <fstream>
    - input and output file stream class
  - #include <sstream>
    - Input and output string stream class



C++ Process Structures

- Sequential
- Decision
  - IF...THEN
  - IF...THEN...ELSE
- Switch
- Looping
  - WHILE
  - DO...WHILE
  - FOR