

- we have also mentioned a few data members, though all of these are actually constants and so are treated somewhat different from data *variables*:
  - string::npos
  - ios::in, ios::out these belong to the ios class (ifstream and ofstream are created based on the ios class)
- the syntax with the class name followed by two colons (::) is used to indicate which class the member after the two colons belongs to. for example:
  - string::npos string is the name of the class and npos is the name of the constant data member belonging to that class
  - ios::in ios is the name of the class and in is the name of the constant data member belonging to that class
  - ios::out ios is the name of the class and out is the name of the constant data member belonging to that class

- we use these classes by declaring variables whose data type is one of these classes, e.g.: string x;
- we call x an *object* of type string
- then we can use the string member functions to operate on the object x, e.g.:
  - string x; x.clear(); x.insert( 0, "hello" );

notice the x. ("x dot") notation

cisc1110-fall2010-sklar-leclX.2

cisc1110-fall2010-sklar-lecIX.2

## why are classes useful?

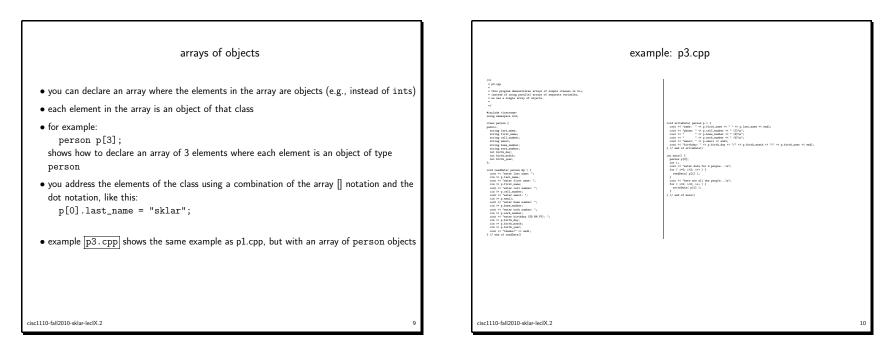
- suppose we wanted to create a program that contains the address book from your cell phone
- look at your cell phone address book:
  - what kind of information is listed for each entry?
  - for example:
    - \* name (first name and last name)
    - \* cell phone number
    - \* email address
    - \* home phone number
    - \* work phone number
- these are called *fields*
- if we wanted to write a program that stored all this information for everyone in our cell phone address book, we could do something like example <u>p1.cpp</u> (we'll pretend we only have 3 friends...)
- cisc1110-fall2010-sklar-leclX.2

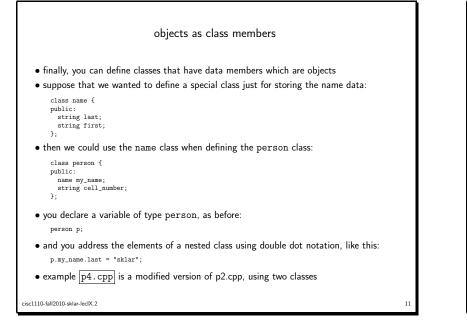
cisc1110-fall2010-sklar-leclX.2

example: p1.cpp	
<pre>/** ** ** ** ** ** ** ** ** ** ** ** **</pre>	
<pre>vert of the state of the s</pre>	
<pre>// in of market // in of</pre>	
cisc1110-fall2010-sklar-leclX.2	6

, ,	<i>iany parallel arrays!</i> so this is why the notion nk together all the fields for each entry in the
<pre>here is a definition of a class that can hold Class person {     public:         string last_name;         string cell_number;         string memail;         string home_number;         string work_number;         int birth_day;         int birth_month;         int birth_year;     }.</pre>	such an entry: - things to notice: * two new C++ keywords: class and public * there is a semi-colon at the END OF THE CLASS DEFINITION, after the last curly brace (})
example <u>p2.cpp</u> shows the previous exam (but for only one person—next, we'll show	ple (p1.cpp) re-written using this simple class how to do it with more than one person)

example: p2.cpp	
/* * p2.op * at progra dometrize the un of single classes in Co., the employ is similar to p1.op, but instand of using separate variables, on group the related variables together bits a ringle class.	
•	
finclude clostream? utig innerpros ndc;	
tikas promi ( pakto: artrag fort_asas; artrag tort_asas; artrag to	
**************************************	
which excludes a prime $p \in I$ and $e^{-1}$ have: $e^{-1} = 0$ for an and $e^{-1} = 0$ for a fame $e^{-1}$ and $f$ and $e^{-1}$ have: $e^{-1} = 0$ for a fame $e^{-1} = 0$ for $e^{-1}$ and $e^{-1} = 0$ for a family defined on $e^{-1}$ for $e^{-1}$ and $e^{-1} = 1$ for a family defined on $e^{-1}$ for $e^{-1}$ and $e^{-1} = 1$ for a family defined on $e^{-1}$ for $e^{-1}$ for $e^{-1}$ and $e^{-1} = 1$ for $e^{-1}$ for $e^{-1}$ for $e^{-1}$ for $e^{-1}$ for $e^{-1}$ for $e^{-1}$ and $e^{-1} = 1$ for $e^{-1}$ for $e$	
is a said ( ) method ( ) writedad ( ); // four of dad()	
cisc1110-fall2010-sklar-leclX.2	8





example: p4.cpp	
/•• • p6.cqp	
* this program demonstrates the use of object members within simple classes in C++. * */	
Flacials (latrua) utig zampice std;	
chan nun ( palici antraginer: bi	
chas pros 4 pros 4 p	
<pre>&gt;&gt; &gt;&gt; &gt;&gt;&gt; &gt;</pre>	
val eritekari penas p ) { erit (* 'uaa (* 'o pagaad frag (* 'o ( pagaad hat (* end)) erit (* 'uaa (* 'o ( pagaad frag (* 'n ()u')) erit (* ' 'o ( pagaad frag (* 'n ()u')) erit (* 'o ( pagaad frag (* 'n ()u')) erit (* 'o ( pagaad frag (* '))) (* 'o ( pagaad frag (* ')))) (* 'o ( pagaad frag (* '))) (* 'o ( pagaad frag (* ')))) (* 'o ( pagaad frag (* '))))) (* 'o ( pagaad frag (* '))))) (* 'o ( pagaad frag (* ')))))))))))))))))))))))))))))))))))	
ina mantol ( perenta pi conditata ( p ); ) // end of man())	
cisc1110-fall2010-sklar-leclX.2	12