

Topics Covered

1. Basic Input/Output
2. Arithmetic in C++
3. Selection/Repetition
4. Functions
5. User defined header file, using files and enumerator datatype
6. Strings
7. Arrays
8. Searching and sorting
9. Structs
10. Testing
Exam 1- Review
3

Some tips – avoiding test anxiety Get a good nights rest I know this is tough, but you don't think as well without sleep Don't skip a meal before an exam Your brain needs protein → try not to eat a high carb meal Don't Cram! Pace your studying Try not to put it off until the last minute If you pace yourself \rightarrow you will be prepared Study with classmates so you can compare notes don't discuss the exam just before coming in their anxiety may impact you Take deep breaths → relax yourself Thing positive thoughts → remind yourself that you are prepared Don't get bogged down on a question answer the questions you know quickly \rightarrow go back to the others **Ask Questions** Calm yourself before you come in... Avoid being late

T/F A C++ identifier may not begin with a digit or an underscore. What is a literal constant? An actual value What type of loop should be used for a program segment that should sum a list of positive integers (it is unknown how many inputs the user will a What should the LCV be? the input number What should the sentinel value be? a negative input When should you initialize the LCV, for each loop? do while - at the top of the loop for loop - in the loop statement When should you update the LCV, for each loop? While loop - at the end of the loop do while - at the top of the loop for loop - in the loop statement

o What is this symbol <<?
insertion operator

O What can be on the left side of the insertion operator?
Left → files, cout

o What can be on the right side of the insertion operator?
right → expressions, variables, constants, literals

o Which manipulator(s) would you use output in columns?
setw()

o What are the escape sequences?
\(\lambda \), \(\lambda \), \(\lambda \), \(\lambda \)

which output manipulators would you use to format floating point numbers?
fixed
setprecision(n)
showpoint
o Know how they work together

```
O What is this symbol >>?
extraction operator

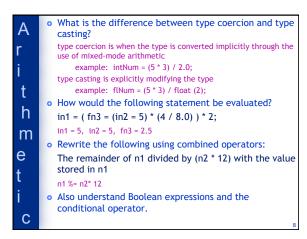
What can appear on the left side of the extraction operator?
cin, files

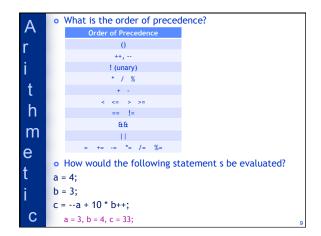
What can appear on the right side of the extraction operator?
variables

When do you have to use .ignore() when reading in c-strings or strings?
-in between the extraction operator and a getline or a .get
-after a .get

What function do you use to get a line of characters and place it in a string
getline(fileName or cin, stringName)

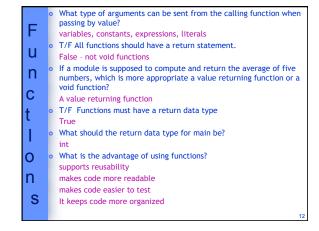
T/F cin always reads directly in from the keyboard
```

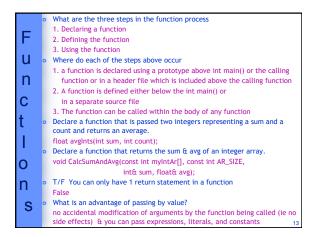


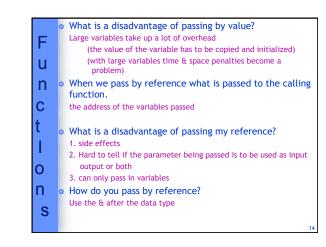


	T/F If a function contains a statement that changes a value
F	parameter, only the copy of the argument is changed, not the original.
	True
u	o How do you declare a function?
n	prototypeWhere does the prototype go?
C	above the int main()
	What is the scope of a local identifier?
t	the end of the block of code in which it was declared
i	 T/F The scope of a value parameter is identical to the scope of a local variable declared in the outermost block of the function body.
	True
0	 T/F In C++, corresponding arguments from a calling function and parameters from a called function must have the same name.
n	False
	When would you use a void function?
S	when nothing is being returned from the function
	or if you need to return multiple values from the function

	c	How do you return multiple values from a function?
F		by passing by reference
	c	What is a side effect?
u		when a variable is passed by reference and is inadvertently changed by the function being called
n	c	How can we avoid side effects?
		1. pass by constant reference
C		2. pass by value
	o c	Parameter passage by reference is used if a parameter's data flow is
l t		a) one-way, into the function
		b) one-way, out of the function
1		c) two-way, into and out of the function
		b & c
O	o c	T/F When passing by value data flow is one-way - into the function
n		True
11	c	What type of arguments can be sent from the calling function when
0		passing by reference?
3	'	variables
		11







T/F An individual component of an array cannot be passed as an argument to a function. The entire array must be sent. False
 How many different data types can you have in one array? none - they all must be the same
 T/F Given the declaration int someAr[20]; int someAr[20]; cout << someAr[3]; outputs the 3rd element in the array. false cin >> someAr[20]; will produce a compiler error false someAr = someAr2; will transfer all values from someAr2 to someAr false - why?

You can return more than one value.

Less overhead, so it is faster and takes less memory.

o Which of the following is true about an array?

a) Arrays are always passed by reference.
b) The name of an array is the address in memory of the first element.
c) Array subscripts always begin at 0.
All of the above

o How would you declare an array of 20 c-strings that can hold up to 11 characters?
char chAr[20][12];
o How would you compare an array of int.
using a for loop → compared each index.

What is the advantage of passing by reference?

What will this statement do: int item[5]={2,12,1}; declare an array called item of 5 elements and assign the values 2, 12, 1, 0, 0 to the elements in the array
 What will this statement do: int item[5]={0}; declare an array called item of 5 elements and assign initialize all the values of the array to 0
 What will this statement do: int item[5]={2,12,1,2,9,5}; give a compiler error
 T/F The compiler will give you an out of bounds error when using arrays if your index is too big or too small. FALSE
 What is stored in the array variable? (e.g. myArray) The base address
 What is the base address? the address of the first element of the array

```
float gpa[5]; // an array holding 5 grade point averages - INP.& OUT.
        // search an array
                                          Know how to use
        index = 0:
       while(index < MAX_ITEMS && !found
                                          loops with arrays
               if (item[index] == searchItem)
                                                 When do you use a for
R
                                                         loop →
                  found=true;
               else
                                                   When do you use a
                                                       while loop?
                  index++:
                                                    Know how to use
                                                   a loop to initialize
       // output the contents of the array
                                                     an array too!
 S
           cout << "\n\nStudent Grade Point Averages"
           for(int i = 0; i < 5; i++)
             cout << "\nGPA for student " << j+1 <<": " << gpa[j];
```

```
o T/F Arrays can be returned as a return value in a function.

False
o T/F Arrays must be passed by reference using the &.
False
o T/F When you pass an array you don't have to include the size in the parameter list for the first dimension.

True
o How should you pass an array when you don't intend to modify it in the function being called.
as a constant
o T/F C-Strings are special arrays.

True
o T/F char name[16]= "Pete"; ⇔ char name[]= "Pete";
False
o Make sure you understand parallel arrays!
```

```
Which loop and conditions to use for?
#1 - Sum an array
For Loop, 0 to AR_SIZE - 1

#2 - Read from an input file into an array
While Loop, InFile && index < AR_SIZE

#3 - Search for multiple instances of an int
For Loop, 0 to AR_SIZE - 1

#4 - Search for one instance of an int
While Loop, !found && index < AR_SIZE

#5 - Output array contents
For Loop, 0 to AR_SIZE - 1

$ #6 - Read from a user into an array
While Loop, input != EXIT && index < AR_SIZE

#7 - Initialize an array to the value -1
For Loop, 0 to AR_SIZE - 1
```

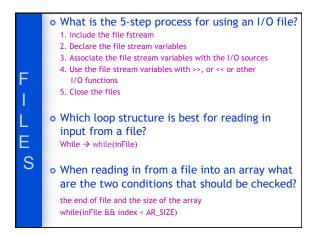
```
How would you declare parallel arrays that could contain a movie
          title, genre, and running time (in minutes)? Assume a const
          AR SIZE
          string title[AR_SIZE];
          string genre[AR_SIZE];
          int time[AR_SIZE];
          How would your read these values in from a file (assume a list
          with \n after each entry) - use the fstream variable inFile?
          index = 0:
          while (inFile && index < AR_SIZE)
a
                getline(inFile, title[index]);
                getline(inFile, genre[index]);
S
               inFile >> time[index];
               inFile.ignore(100, '\n');
               index++:
        Know how to use files.
```

```
o How would you declare a multi-dimensional array that would hold 10 scores for 5 people?
int scores[5][10];

o How would you read these values into the array from a file (assume the infile variable is already assigned - assume 10 scores per row)?
int row;
int col;
row = 0;
while (inFile && row < 5)
{

for (col = 0; col < 10; col++)
{
    inFile >> scores[row][col];
}
row ++;
}
o Know how to output multi-dimensional arrays and how to initialize them.
Exam 2 - Review
```

```
o How would you compare C-Strings.
strcmp
o T/F Strings have null terminators.
False
o T/F String size is dynamically allocated.
True
o T/F You can't use getline with a string only with C-strings.
False
o If we declare 3 strings. This is valid str3 = str1+str2.
yes
o What will it do?
concatenates the strings
str1 = "abc";
str2 = "def";
str3 will = "abcdef";
```



```
T/F You can put executable code in a header file.
           False

    T/F It doesn't matter what order the directives and declarations

е
           appear in a header file.
а
           False
        • What order should the directives appear and why?
d
           typedefs /enum types
е
           constants
           function prototypes
          What are these lines in a header file for?
ø
           #ifndef MYHEADER_H_
           #define MYHEADER H
           #endif
           they prevent you from replicating definitions
        Does the extension for your header file matter?
 е
           Yes! - it should be .h.

    How do you include a user-defined header file?

 S
           #include "myheader.h"
```

```
What are the 3 basic categories of datatypes?
           simple
           structured
u
           address
m
        o What are the 3 basic simple datatypes
s
           integral → char, short, int, long, & bool
           floating → float, double
           enum → user defined
        • What category do arrays fall under?
           structured (which also includes structs, unions, and classes)
р
        o How would you define an enumerated type to represent the
е
d
           enum Seasons {SPRING, SUMMER, WINTER, FALL};
е
           Why would you use enumerated types?
           they make your code self-documenting \Rightarrow increase readability
           simplifies coding
```

```
o Since enums are essentially evaluated into integers, can you
           perform arithmetic on them?
u
           only if you typecast or assign the result into an integer
m
        • How would your output an enum?
s
           switch statement (know how to do this)
        • What does a typedef do?
           creates an additional name for an already existing data type
        Can you do this?
р
           typedef float FloatArrayType[100];
е
d
        o How would you use it?
           FloatArrayType myArray;
е
```

```
o What are the advantages of a sequential search?
easy to implement
array doesn't have to be sorted

o What is the disadvantage of a sequential search?
it can be inefficient

what is the advantage of a binary search?
more efficient

when would you use a sequential search over a binary search?
when you don't have to search often AND the list is unsorted
⇒ binary searches have the overhead of sorting the list first

o Know how to identify the searching and sorting algorithms.
```

```
Sorting Algorithms
             Buddle Sort
                           Selection
                           Sort
                                         Sort
Outer Loop
             For Loop
                           For Loop
                                         For Loop
Outer Loop
                                         Index 0 to
             Index 0 to
                           Index
                           AR_SIZE - 1
Condition
             AR_SIZE - 1
                                         AR_SIZE - 1
                           to 0
Inter Loop
             For Loop
                           For Loop
                                         While Loop
Main Action
                           Select the
                                         Insert each
             Swap
             adjacent
                                         number in
                           largest
                           number for
             numbers in
                                         their correct
             their correct
                           each loop
                                         location
             order
```

```
o What is the advantage of using structs?
easier to organize related data items.

o What is a member?
a field within the structure
Are aggregate operations allowed on structs?
only assignments
Can you pass structs by value or reference?
yes
Can structs be a return type?
yes
```

```
• Define a struct called DvdRec, that contains the title,
          genre, and running time.
           struct DvdRec
              string title;
              string genre;
u
       o Declare an array 100 elements of that struct called
          movies.
C
          DvdRec movies[100];
       • How would you output the title of the 10<sup>th</sup> element in
          your array?
 S
          cout << movies[9].title;</pre>
        • Be able to write a function that can read into an array
          of structs or output an array of structs.
```