Final Exam

Due Dec 14 at 11:59pm **Points** 100 **Questions** 35 **Available** Dec 14 at 7am - Dec 14 at 11:59pm about 17 hours
Time Limit 120 Minutes

Instructions

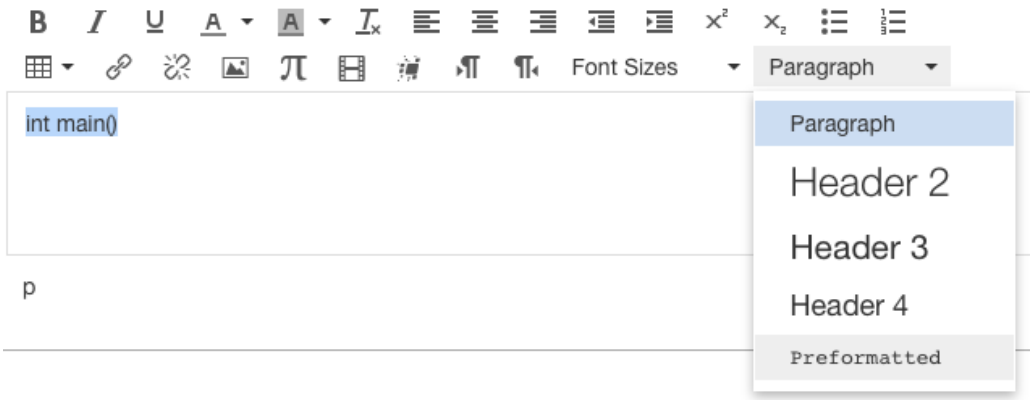
This is a final examination of your understanding of the material covered in our labs, lectures, and readings, covering Modules 3-13, but focusing on Modules 7-13. It is worth 100 "exam points" -- half of the exam score for the course, meaning it amounts to 5% of your final grade.

There are true/false, multiple choice, short essay, and fill-in-the-blank questions. The exam is *open book, open note, and open internet,* but "closed neighbor" and email, texting, and any outside contact is strictly prohibited.

There is a question about the end-of-semester survey. If you have not yet taken it, here's the URL: **www.surveymonkey.com/r/fa2016comscPost (http://www.surveymonkey.com/r/fa2016comscPost)** . You may want to do that before starting the exam!

You have 120 minutes from the time you start the exam! If you pause or leave to come back later, the timer will still run. It will NOT pause.

For questions that ask you to write code, it will look a lot better and be easier for you to follow what you're typing if you write in "preformatted" instead of the default "paragraph". Switch like this:



Once you start the exam, you'll have ***120 minutes of clock time*** to complete it. If you need to pause your exam, simply navigate to the Home page and resume later. But note that the *time period allowed for the exam remains the same*.

Once you choose to Submit the exam, you're done. You may see a partial score including just the questions that Canvas can score automatically. Your actual score will be revealed once the exam is graded by the instructor.

This quiz was locked Dec 14 at 11:59pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	82 minutes	82 out of 100

Score for this quiz: **82** out of 100
Submitted Dec 14 at 10:41pm
This attempt took 82 minutes.

Question 1 — 18 / 20 pts

Identify the elements in the C++ code:

```
template <typename V>
class SomethingOrOther
{
  ...
  V doSomething(const V&);
  V doSomethingElse() const {return V( );}
};

template <typename V>
V doSomething(const V& value)
{
  V result;
  ...
  return result;
}

int main()
{
  SomethingOrOther<string> a;
  string aString;
  ...
  cout << a.doSomething(aString);
}
```

Correct! **member function** doSomething

Correct! **ADT** typename V

Correct! **host object** a

Correct! **parameter object** value

You Answered **templated class** template <typename V>

Correct Answer SomethingOrOther

Correct!	**returned object**	result
Correct!	**inline function**	doSomethingElse
Correct!	**default data type value**	V( )
Correct!	**setter member**	doSomething
Correct!	**local variable**	aString

Other Incorrect Match Options:
- int main()
- template <typename V>

Question 2

2 / 2 pts

What is the purpose of the "trailing const"?

Your Answer:

It represents getter function (not mutable)

Question 3

0 / 2 pts

Explain what is a "hole"?

Your Answer:

memory leak

unused index in an array

Question 4

0 / 2 pts

Why do we write both an operator square-bracket setter *and* a getter?

Your Answer:

Setter function can not be used in getter and getter can not be used in setter. So If we write both an operator square-bracket, compiler can choose one of them.

Getter is for use with const versions of the data structure

Question 5

2 / 2 pts

What's the purpose of the "ifndef test"?

Your Answer:

The purpose of ifndef test is to confirm that the #ifndef, #define, and #endif statements are written correctly. If ifndef test were failed, program would make compiler error when cpp file has the same header file included more than once

Question 6

2 / 2 pts

What's the purpose of the object copy test?

Your Answer:

Array<int> d = a;
Check that the object is copied well.

d and a have same values, but they have different addresses.

Question 7

2 / 2 pts

What's the purpose of the object assignment test?

Your Answer:

Array<int> e; e = a;

Check whether object assignment worked well.

object assignment is copy construct + deallocated the existing array before the new one can take its place

Question 8

2 / 2 pts

What's the purpose of const object testing?

Your Answer:

The purpose of this test is confirm that the public member functions that we intended to be getters are in fact coded as getters

Question 9

0 / 2 pts

What's the purpose of timing tests?

Your Answer:

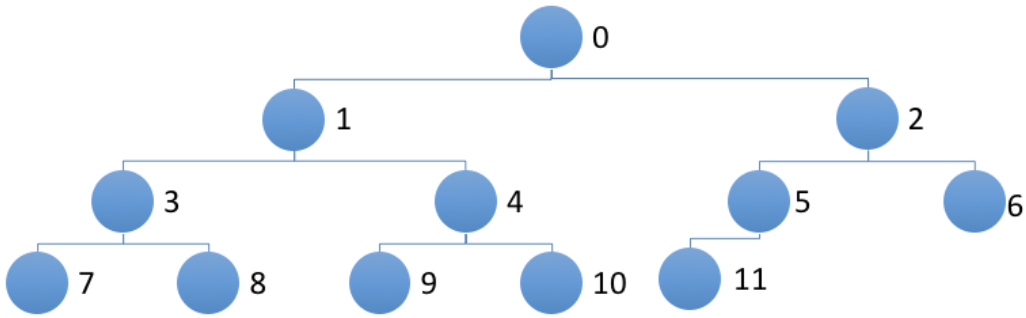
compare between expected runtime and real runtime.

confirm big oh

Question 10

2 / 2 pts

Write the contents of the following tree in ***postorder*** sequence, single-space separated:



Correct!

7 8 3 9 10 4 1 11 5 6 2 0

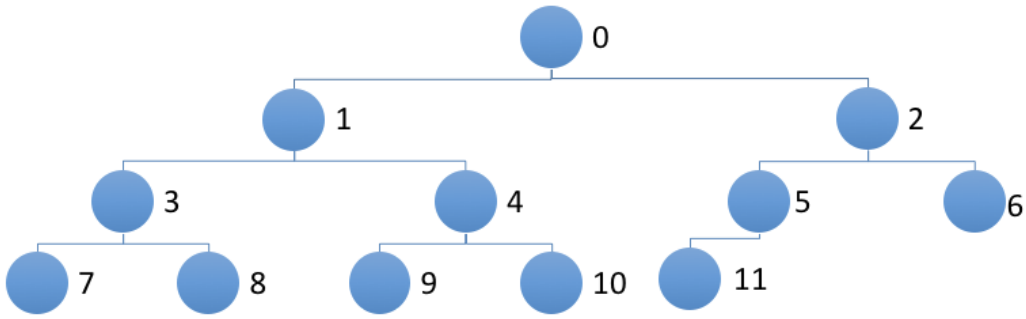
Correct Answers

7 8 3 9 10 4 1 11 5 6 2 0

Question 11

2 / 2 pts

Write the contents of the following tree in ***preorder*** sequence, single-space separated:



Correct!

0 1 3 7 8 4 9 10 2 5 11 6

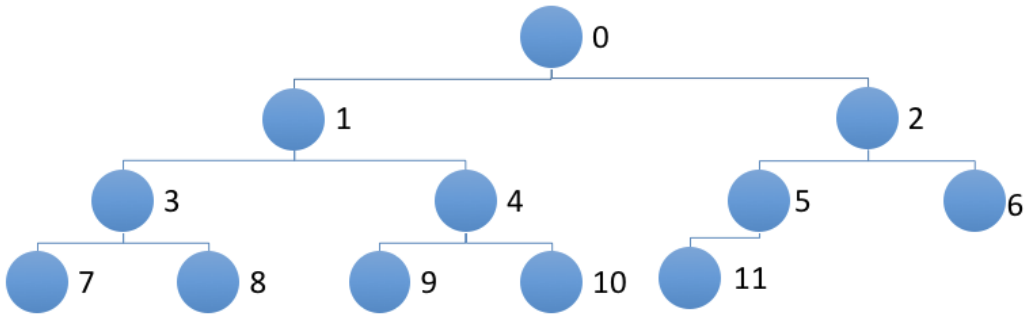
Correct Answers

0 1 3 7 8 4 9 10 2 5 11 6

Question 12

0 / 2 pts

Write the contents of the following tree in ***inorder*** sequence, single-space separated:



You Answered

7 3 8 1 9 4 10 11 5 2 6

Correct Answers

7 3 8 1 9 4 10 0 11 5 2 6

Question 13

6 / 6 pts

Match the following tree interation sequences with their acronyms:

Correct! **inorder** — LVR

Correct! **preorder** — VLR

Correct! **postorder** — LRV

Other Incorrect Match Options:

- RVL
- RLV
- VRL

Question 14

2 / 2 pts

Name the 2 common search methods for iterating over graphs, to find all vertices reachable from a start vertex. Write their full names ***and*** 3-letter abbreviations.

Your Answer:

BFS - Breath First Search
DFS - Depth First Search

Question 15

3 / 3 pts

Objects stored in an STL priority_queue ***must*** have defined for them: *(click all that apply)*

- `operator<=`
- `operator==`

Correct!

- [x] `operator<`

Question 16

0 / 2 pts

Hash code values that are returned by hashing functions can be:

Correct Answer

- ( ) any value in the int range, including negative values

You Answered

- (•) any value from 0 to one less than the capacity of the hash table in which it is used

- ( ) any non-negative int value

Question 17

0 / 2 pts

Write a C++ *code block* (not a *function*) for use in hash table functions to convert a hash code for an object "**a**" to a wrapped index named **wrappedIndex**. Assume that the hash code function has the prototype **int hashCode(const K&)** and that the hash table capacity is **CAP**.

Your Answer:

Question 18

2 / 2 pts

In order to maintain O(1) for hash table operations, how full should the hash table be allowed to get before expanding its capacity (approximately)?

Using probing: 50 %

Using chaining: 80 %

Answer 1:

Correct!

50

Answer 2:

Correct!

80

Question 19

2 / 2 pts

What's the big oh for *successful* linear search of an unsorted linked list?

Correct!

- (•) O(n)
- ( ) none of these
- ( ) O(log n)
- ( ) O(1)

Question 20

2 / 2 pts

What's the big oh for *unsuccessful* linear search of a sorted array?

- ( ) none of these
- ( ) O(log n)

Correct!

- (•) O(n)
- ( ) O(1)

Question 21

2 / 2 pts

What's the big oh for *unsuccessful* binary search of a sorted array?

- ( ) O(1)
- ( ) none of these

Correct!

- (•) O(log n)

O(n)

Question 22

2 / 2 pts

What's the big oh for the insertion sort algorithm when applied to a *randomly arranged* array?

Correct!

- (•) $O(n^2)$
- ( ) O(n log n)
- ( ) none of these
- ( ) O(log n)

Question 23

2 / 2 pts

What's the big oh for the insertion sort algorithm when applied to an *already arranged* array?

- ( ) O(log n)
- ( ) none of these
- ( ) O(n log n)

Correct!

- (•) O(n)
- ( ) $O(n^2)$

Question 24

0 / 2 pts

The Shell sort algorithm runs several steps before ending in a regular insertion sort as its last step. Yet it's generally faster than a regular insertion sort. Explain how that can be true.

Your Answer:

If we use shell sort with dividing 3, each short array's sorting takes $^1/_9$ the runtime of the whole array, because the size is $^1/_3$
Big O of shell sort is $n^{1.25}$ and Big O of insertion sort is n2

Question 25

2 / 2 pts

What is the big oh for the Shell sort? O (n 1.25)

Answer 1:

Correct!

1.25

Question 26

0 / 2 pts

What's the big oh for an array-based stack push operation?

- O(log n)
- none of these

You Answered

- O(n)

- O(1)

Question 27

2 / 2 pts

What's the big oh for a linked-list-based stack push operation?

- O(log n)
- none of these

Correct!

- O(1)

- O(n)

Question 28

2 / 2 pts

What's the big oh for a linked-list-based queue push operation, when the queue maintains both a head and a tail pointer?

- O(log n)
- none of these
- O(n)
- O(1)

Correct!

Question 29

2 / 2 pts

What's the big oh for a linked-list-based queue push operation that adds to the end of the list, when the queue has *no* tail pointer?

- O(log n)
- O(n)
- none of these
- O(1)

Correct!

Question 30

2 / 2 pts

What's the big oh for a linked-list-based stack **size( )** getter function, when the only member data item in the stack is a head pointer?

- O(n)
- O(1)

Correct!

- O(log n)
- none of these

Question 31

2 / 2 pts

What's the big oh for a radix (also known as “bogo”) sort?

- $O(n^2)$
- O(n) — Correct!
- none of these
- O(1)

Question 32

2 / 2 pts

I have my own version of the DynamicArray template that adds three functions: sort, linear search, and binary search. I have an idea to combine the linear search and binary search functions into a single search function that can detect if the array is already in order – if so, it will apply a O(log n) binary search algorithm, otherwise it will do a O(n) linear search. To perform this detection, the function will run a for-loop to see if the array is already in order. Does this sound like a good idea? Explain.

Your Answer:

Bad idea. When we detect if the array is already in order, this step is already Big O(n). So It is not efficient.

Question 33

2 / 2 pts

Explain how “associative arrays” are different from regular arrays:

Your Answer:

In associative arrays, Index can be anything, not just number.

Question 34

10 / 10 pts

Match the most appropriate data structure to an application:

Correct! **You're simulating a buffered keyboard and would like to process the keystrokes and then output them to the console in the order they were typed.** — queue

Correct! **Your simulation processes service events in order, based on their completion time. But they get added to the simulation in order by their start time, and not by their completion time.** — priority queue

Correct! **You're writing a program that iterates over a graph and you would like to use a separate additional data structure to keep track of whether a vertex has been visited or not.** — associative array

Correct! **You're writing expression parsing program for an early HP calculator that uses reverse polish notation.** — stack

Correct! **You want to represent tasks in a project plan, in order to determine the "critical path".** — graph

Question 35

1 / 1 pts

Did you complete the end-of-semester survey whose link is on our syllabus? If not, click **HERE** (http://www.surveymonkey.com/s/sp2016comscPost) to complete it now, and then answer below. Note that the survey is anonymous.

Correct! (•) Yes

( ) No

Thanks for taking my class!

Quiz Score: **82** out of 100