

This exam is Copyrighted by Dr. John Gibbons of the University of Kansas. DO NOT duplicate or redistribute. No outside assistance (e.g. other students, Chegg.com, prior exams) is permitted.

EECS 268 2021 Fall Midterm

Rules

- **Time limit:** 50 minutes
- Only mark your answers within designated answer boxes
- Read and sign below
- **Unauthorized aid:** Other students, notes, books, phones, any electronic devices, any material not provided to you on the exam (other than your writing utensil)
- **Authorized aid:** Your amazing brain

Your Name	MORGAN BERGEN
KUID	M358 B583

EECS ~~268~~ 268
MWF 1 PM

This exam is Copyrighted by Dr. John Gibbons of the University of Kansas. DO NOT duplicate or redistribute. No outside assistance (e.g. other students, Chegg.com, prior exams) is permitted.

[1pt] Honesty agreement

I'd like this exam to still be an assessment of your skill and understanding, and not your ability to find answers online. Please type the name of someone who would be heartbroken if they knew you cheated on this exam.

Their Name	MORGAN BERGEN
------------	---------------

This exam is Copyrighted by Dr. John Gibbons of the University of Kansas. DO NOT duplicate or redistribute. No outside assistance (e.g. other students, Chegg.com, prior exams) is permitted.

[12pts] Pick the right tool

In the following questions, you will be given a description of a problem from various customers. You must pick the data structure that would be best for the given task.

1. [3pts] "Hi! I need help organizing my office hours. If a student arrives before another student, then they should get helped first. Can you help?"

- ☐ a. Stack
- ☒ b. Queue
- ☐ c. List
- ☐ d. None of the above



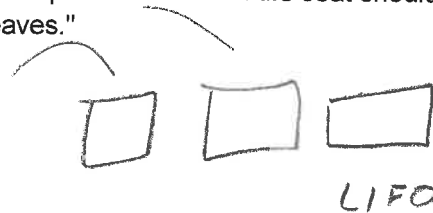
Answer:

B



2. [3pts] "I've received complaints from the passengers on my airplane about people crawling over them to get out of their seats. The seats are arranged window, middle, then aisle. The person in the window seat sits down first and shouldn't get to leave until the person in the middle seat leaves. The person in the middle seat shouldn't get to leave until the person in the aisle seat leaves."

- ☒ a. Stack
- ☐ b. Queue
- ☐ c. List
- ☐ d. None of the above



Answer:

A

3. [3pts] "I'm trying to make an app to help people at the grocery store. I need to be able to add items that I want to buy then mark them off after I add them to my grocery cart."

- ☐ a. Stack
- ☐ b. Queue
- ☒ c. List
- ☐ d. None of the above

Answer:

C

This exam is Copyrighted by Dr. John Gibbons of the University of Kansas. DO NOT duplicate or redistribute. No outside assistance (e.g. other students, Chegg.com, prior exams) is permitted.

4. [3pts] "Heya, I work at a checkout at a grocery store and I have a problem with my customers. All these idiots are too busy staring at some kind of grocery store app instead of looking to see if they just cut in line. Can you help me make sure the people who come to the checkout late don't cut in line?"
- a. Stack
 - ☒ b. Queue
 - c. List
 - d. None of the above

Answer:

B

[27pts] Conceptual

1. [3pts] When does the type variable of a templated class (e.g. the T in template <typename T>) become well defined? Choose one.
- A. When you include the header file in another file
 - ☒ B. It is never defined
 - C. When an object of that type is created (having its constructor called)
 - D. None of the above

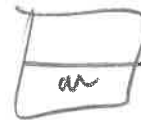
Answer:

C

<CHAD

2. [3pts] If a function named *func* declares an integer locally, when is that integer deallocated? (Choose one)
- A. When the function returns
 - ☒ B. When main returns
 - C. When delete is called on the variable
 - D. None of the above

↑
MAIN?



Answer:

B

3. [3pts] If a function named *func* declares a Circle* named *ptr*, when is that pointer (not anything it's pointing to, but the pointer) deallocated?
- A. When main returns
 - B. When the function returns
 - ☒ C. When delete is called on a pointer to the object
 - D. None of the above

CIRCLE* PTR = NULL



DELETE

Answer:

D

This exam is Copyrighted by Dr. John Gibbons of the University of Kansas. DO NOT duplicate or redistribute. No outside assistance (e.g. other students, Chegg.com, prior exams) is permitted.

4. [3pts] Why shouldn't `getEntry` be in charge of catching and handling the exception it throws? Choose one.

- ☐ A. Because the operating system catches exceptions, not our code
- ☐ B. Because Exceptions are automatically thrown when memory errors occur
- ☒ C. Because if `getEntry` catches the exception it throws, it still needs to return a value.
- ☐ D. None of the above

Answer:	C
---------	---

T GET ENTRY
C RETURN(ENTRY)

5. [3pts] What happens if a thrown exception is never handled? Choose one.

- ☐ A. Nothing
- ☒ B. Program ends due to unhandled exception
- ☐ C. Memory error
- ☐ D. None of the above

Answer:	B
---------	---

6. [3pts] Where are new values inserted into our lists?

- ☒ A. At the front
- ☐ B. At the back
- ☐ C. List do not allow new values to be added
- ☐ D. None of the above

Answer:	A
---------	---

7. [3pts] By default, assignment operators and copy constructors create what kind of copy (deep or shallow)?

- ☒ A. Shallow
- ☐ B. Deep
- ☐ C. None of the above

Answer:	A
---------	---

This exam is Copyrighted by Dr. John Gibbons of the University of Kansas. DO NOT duplicate or redistribute. No outside assistance (e.g. other students, Chegg.com, prior exams) is permitted.

8. [3pts] When is a copy constructor called?
- A. When an object is passed by value
 - B. When you declare an array of objects
 - C. When you save an object to file
 - ☒ D. None of the above

Answer:

D

9. [3pts] The ability of a base type pointer or reference to run a method created in a derived type is called: (Choose one)
- A. Inheritance
 - B. Derivation
 - ☒ C. Polymorphism
 - D. None of the above

SHAPE* PTR → AREA()

Answer:

C

This exam is Copyrighted by Dr. John Gibbons of the University of Kansas. DO NOT duplicate or redistribute. No outside assistance (e.g. other students, Chegg.com, prior exams) is permitted.

[10pts] Inheritance

Below are the descriptions of how various classes are related to each other.

Descriptions:

- ☒ Shark is a Fish
- ☒ Corgi is a Dog
- ☒ Bird is an Animal
- ☒ Tabby is a Cat
- ☒ Dog is an Animal
- ☐ Duck is a Bird
- ☒ Cat is an Animal
- ☒ Fish is an Animal

Assume a class hierarchy of these classes is implemented. Now, answer the questions below regarding whether or not each operation would be legal or illegal.

Action	Legal or illegal?
Shark pointer pointing to a Fish object	ILLEGAL
Using an Animal reference to call a method <i>doTrick</i> that was not defined until the Dog class	ILLEGAL
Using a Cat pointer to call an Animal method	LEGAL
Using a Bird pointer to point to a Duck object	LEGAL
Passing a Shark to a function that takes Animals by reference	LEGAL
Passing an Animal to a function that takes Ducks by reference	ILLEGAL
Passing a Tabby object to a function that takes Fish by reference	ILLEGAL
Adding a method to the Cat class that takes Fish by reference	ILLEGAL
Dog class redefining a method that was already defined in the Animal class	LEGAL
Animal pointer to a Corgi object	LEGAL

SHARK → FISH

DOG → ANIMAL ()

CAT → ANIMAL

BIRD → DUCK

SHARK (ANIMAL)

DUCKS (ANIMAL)

DUCK YELLOW;
BIRD* BROWN → YE

This exam is Copyrighted by Dr. John Gibbons of the University of Kansas. DO NOT duplicate or redistribute. No outside assistance (e.g. other students, Chegg.com, prior exams) is permitted.

[25pts] Code Writing

Assume you are in Queue.cpp. The Queue class has an m_front and m_back both of which are of type Node<T>* and are set to nullptr in the constructor. Also assume that your nodes have a T m_entry and Node<T>* m_next members and have getters and setters for both. Below you will implement dequeue. There are no other member variables, but you may assume all other Queue methods are implementing and working (enqueue, peek_front, isEmpty). A std::runtime_error with the message "Queue empty" should be thrown when trying to dequeue an empty queue.

```
template <typename T>
void Queue<T>::dequeue() {
    //your code below
    IF ( m_FRONT == NULLPTR ) {
        THROW (STD:: RUNTIMEERROR ("QUEUE EMPTY"));
    } ELSE {
        NODE <T> * TEMP = NULLPTR;
        TEMP = m_BACK;
        DO {
            TEMP -> SETNEXT;
        } WHILE ( TEMP -> GETNEXT == m_FRONT )

        TEMP = m_FRONT
        DELETE TEMP;
        TEMP -> NULLPTR;
    }
}
```


This exam is Copyrighted by Dr. John Gibbons of the University of Kansas. DO NOT duplicate or redistribute. No outside assistance (e.g. other students, Chegg.com, prior exams) is permitted.

[25pts] Code Writing

Assume you are implementing a LinkedList class. LinkedList has two private members:

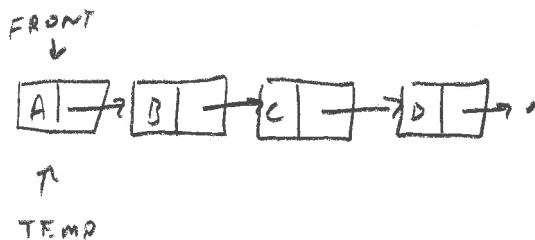
- Node<T>* m_front
- int m_length

You may assume that all other Linked List methods are working (i.e. insert(index, entry), remove(index), getEntry(index), length(), clear())Also assume that your nodes have a T m_entry and Node<T>* m_next members and have getters and setters for both.

You will define a method called *reverse*. This method will keep all the values in the list, but will reserve the order they are in. Make sure to adjust **all pointers** that need updating!

Original list	After reverse is called
A, B, C, D	D, C, B, A
A	A
A, B	B, A
<empty list>	<empty list>

You will write your code in the
box on the following page



This exam is Copyrighted by Dr. John Gibbons of the University of Kansas. DO NOT duplicate or redistribute. No outside assistance (e.g. other students, Chegg.com, prior exams) is permitted.

```
template <typename T>
void LinkedList<T>::reverse() {
    //your code below

    NODE * TRAVERSE = NULLPTR;
    TRAVERSE = M_FRONT;
    FOR (INT i = 0; i < 4; i++) {
        CHAR ENTRY;
        ENTRY = TEMP -> GET_NEXT
        TEMP -> SET_NEXT(ENTRY);
        M_FRONT -> SET(ENTRY);
    }
    DELETE TEMP;
    TEMP -> NULLPTR;
```