MIDTERM EXAM INSTRUCTIONS

1. Midterm Exam: **25 points w/ 0 E.C. points**
2. Due Date & Time: **04-25-2021 at 05:00 AM**

WHAT TO SUBMIT

1. Take-home Exam Report, 1 PDF

HOW TO SUBMIT AND THE RULES TO FOLLOW

- Submit via iLearn, the Assignment and Exam Submission section
- Please follow the exam instructions
- Please follow the Course Policy on Student Conduct and Academic Honesty

PERFORMANCE TRACKER		
ASMT	**GRADE**	**YOUR GRADE**
ZOOM	05	
01	15	
02	100	
03	100	
MIDTERM 01	25	
04-PREPARATION	25	
04	75	
MIDTERM 02	25	
TOTAL	370	

A: 90-100% **B**: 80-89% **C**: 70-79% **D**: 60-69% **F**: 0-60%
The course grader provides feedback to your assignments on iLearn.

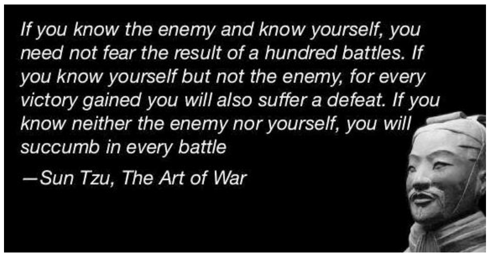
ABOUT

The goal of this take-home exam is for us to **know what we do not know**.

We are taking this exam as seriously as how we take an actual exam in class. Please,

1. Follow all the rules and the guidelines listed at the top of page 1 and page 2
2. Read each question carefully before answering

We will go over the answers to all the exam questions together in class.



If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle

—Sun Tzu, The Art of War

STEP A – Take the Exam, **10 points**

1. Allocate 50 quiet minutes to take the exam on page 2 to the last page.
2. Record the date and time when you start.
3. Stop right at minute 50. Record the date and time when you stop the exam.

EXAM
Full Name | SFSU ID
- Each question
    - Answer, Step A
    - *Corrected Answer, Step B*
    - *More practice, Step C*
    - *Other notes*

- Next question

STEP B – Correct Your Answers, **10 points**

1. Review the related course materials and write code when necessary to find a correct answer for each question. We should be able to find all the answers using the packages, the in-class discussions, our assignments, and the other course materials.
2. At the end of each of your oringal answers, type in *italic* text and:
    - Give your orginal answer a score.
    - List all the mistakes then explain why, you think, you made the mistakes. Add the correct answer you found. Document how you found the correct answer. Document where you found the materials which support the answer.
    - If you did not make any mistakes, please document how you verified that your answer was accurate. Document how you found the correct answer. Document where you found the materials which support the answer. Outline how you could have done better.
    - Record your total score out of 100 points for all the orginal answers.

STEP C – Reflect and Retake the Exam, **5 points**

1. **Problem Solving**: Reflect if you managed the exam time efficiently and if you strategized your test-taking successfully.
2. Repeat steps A to C again if necessary. Please keep appending new contents as directed in Step B.2.
3. Think if the same topics will be tested again in our final exam, what questions we may get.

It is a good idea to do every step of this exam thoroughly. We are creating a set of materials which we will use to review for the final exam. And this is also the best way to prepare ourselves to succeed in the last part of the semester. Thank you.

1. Section, Date and Time:
   TIC and TAC, due ##-##-#### at ##:## ##
2. Midterm Exam (2 exams, 0 dropped): 100 points

Full Name in **CAPITAL LETTERS**	**SFSU ID**

3. To prepare for this exam, please review all the related materials including the packages, slides, mock-up exam(s), reading assignments, in-class practices, sample programs posted in the File Manager, and assignments.
4. You do not need to print this exam. No papers. No handwriting. No scanned images. No screenshots. Please type up all your answers in the answer space available in the exam. The provided exam will be in Microsoft Word format. Please submit a single PDF via iLearn.
5. All the rules of an actual exam apply to this exam such as: closed books, closed notes, closed IDEs, and no communication with anyone except the course instructor. The course instructor will be available on Zoom (zoom.ducta.net) or via email during the exam time. You cannot use any other materials or tools but only the provided exam which will be in Microsoft Word format.
6. Please ask all your questions, if any, during the review sessions. Thank you.

HONOR CODE:

- Please follow the CS Department's policies: https://cs.sfsu.edu/student-policies

- Please follow the course's policies: http://csc340.ducta.net/00-README-StudentConduct_AcademicHonesty.pdf

---

PART A – 50 Points

A.1 - 10 pts

Please explain in detail Scott Meyers's point: Use weak_ptr for shared_ptr like pointers that can dangle.

A.2 - 10 pts
How are Smart Pointer functions **move()**, **reset()**, and **release()** different from each other? Please also explain in detail which function is most dangerous and why?

A.3 - 16 Points

```
...
class Name {
public:
	Name() {}

	Name(string name) {
		this->name = name;
	}

	~Name() {
		cout << this->name << ": Destructor called." << endl;
	}

	string getName() const {
		return this->name;
	}
private:
	string name{ "N/A" };
};

void passByMove(const unique_ptr<Name> uPtr_M) {
	cout << "@uPtr_M: " << uPtr_M << endl;
	cout << "getName(): " << uPtr_M->getName() << endl;
}

void passByRef(const unique_ptr<Name>& uPtr_R) {
	cout << "@uPtr_R: " << uPtr_R << endl;
	cout << "getName(): " << uPtr_R->getName() << endl;
}

void passByShare(const shared_ptr<Name> sPtr_S) {
	cout << "@sPtr_S: " << sPtr_S << endl;
	cout << "getName(): " << sPtr_S->getName() << endl;
	cout << "use_count(): " << sPtr_S.use_count() << endl;
}

Name* passByValue(const unique_ptr<Name> uPtr_V) {
	cout << "@uPtr_V: " << uPtr_V << endl;
	cout << "getName(): " << uPtr_V->getName() << endl;
	return uPtr_V.get();
}

int main() {

	cout << passByValue(make_unique<Name>("Goofy")) << endl;

	unique_ptr<Name> uPtr{ make_unique<Name>("Mickey") };

	passByRef(uPtr);
	cout << "name_uPtr: " << uPtr << endl;

	passByMove(move(uPtr));
	cout << "name_uPtr: " << uPtr << endl;

	uPtr = make_unique<Name>("Minnie");
	shared_ptr<Name> sPtr{ uPtr.release() };
	passByShare(sPtr);

	cout << "END of Program" << endl;
	return 0;
}
```

How many lines does this program output? ___________

Please give the **output** of the program. *Use **@A, @B, @C, @D**, and **nullptr** to represent memory addresses.*

01	
02	
03	
04	
05	
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

A.4 - 4 pts
Please explain in detail how to manually destroy an existing Smart Pointer control block.

A.5 - 10 Points

```
…
int funcB(int);

int funcA(int n) {
	if (n <= 1)
		return 217;
	else
		return n + funcB(n - 2);
}

int funcB(int n) {
	if (n <= 2) {
		return 3;
	} else {
		if (n > 4) {
			return n * funcA(n - 5);
		} else {
			return n - funcB(n - 1);
		}
	}
}

int main() {
	cout << funcA(13);
	return 0;
}
```

What is the output of this program? **Please show our work.**

PART B – 40 Points

B.1 - 20 Points

```
...
class Name {
public:
    Name() {}
private:
    string name{ "CS" };
};

shared_ptr<Name> func() {
    unique_ptr<Name> obj{ make_unique<Name>() };
    // #1 Insert Code
}

int main() {
    // #2 Insert Code
}
```

[#1 Insert Code]: **Write code** to keep the object which **obj** owns alive outside of the scope of function **func**. *Hint: The code should also support our task in #2 Insert Code.*

[#2 Insert Code]: **Write code** to test if the object owned by **obj** is alive in the scope of function main. If it is, please output its address. If not, please output “Object destroyed.”

B.2 – The Problem, 5 Points
Similar to what we did in ASMT 4, we are to carry out the three following steps. We use
1. **Copy constructor** to create a new bag using a bag passed in. However, **please note** that the order in which this constructor stores the items is the **reverse** of the original order (in which the items appear in the passed-in bag).
2. Function **addVector,** which takes a vector as the only parameter, to add the entries from the vector to the new bag.
3. Function **removeLastThree**, which takes an item as the only parameter, to remove the last three occurrences of the item in the bag. *It is OK to make assumptions. Please state our assumptions, if any.*

Please explain in detail **how each step works** and **what** the new bag contains after each of the steps is executed.
- Please use linked Nodes diagrams in our answer.
- Please use the data below in our explanation.

1. Passed-in bag:	‘e’, ‘l’, ’e’, ’c’, ’t’, ’r’, ’i’, ’c’, ’a’, ’l’
2. Passed-in vector:	’e’, ’n’, ’g’, ’i’, ’n’, ’e’, ’e’, ’r’
3. Passed-in item:	‘e’

Copy constructor (reverse order)

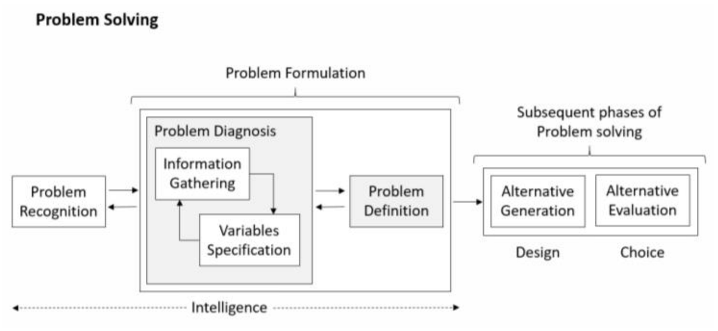
function **addVector**

function **removeLastThree**

B.3 – The solution, 15 Points
Please code the **Copy Constructor** (the reverse) and the **removeLastThree** function described in B2 without using the existing functions.

PART C – 10 Points

C.1 - 5 pts



What did the IDEO team do during the Problem Formulation phase? What was the most important question? Can we use the same approach in Software Development?

C.2 - 5 pts
Your programming team lead/leader asks you to use raw pointer to implement a memory bound function. You know that using Smart Pointers is the way to go. And you heard that the leader does not know Smart Pointers. What would you do and say? Please provide your answer in detail.

PLEASE DO NOT DETACH THIS PAGE FROM YOUR EXAM.
You may get partial credit for what you write on this page.