1. 50 points w/ 0 E.C. points
2. Due Date & Time: **07-11-2021 at 11:55 PM**

Performance Tracker		
Asmt	**Grade**	**Your Grade**
Canvas	05	
01	15	
02	100	
03	100	
Midterm Exam	50	
Total	270	

A: 90-100% **B**: 80-89% **C**: 70-79% **D**: 60-69% **F**: 0-60%

The course grader provides feedback to your assignments on Canvas.

What to submit

1. 1 PDF

How to Submit and the Rules to Follow

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

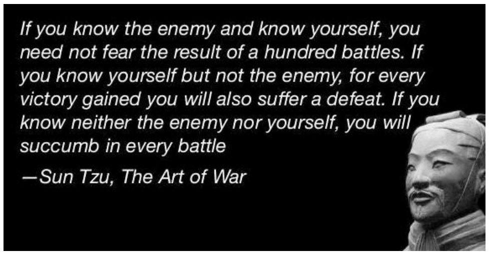
About

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

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 second midterm or 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 second half of the semester. Thank you.

1. Section, Date and Time:
   TIC, TAC, and TOE. Due ##-##-#### at ##:## PM

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

2. [redacted] (1 exam, 0 dropped): 100 points
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 Canvas.
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 – 90 Points

A.1 - 5 pts - How is a **pointer** different from a **reference variable**? And please give a code example.

Would these statements cause an error? Why or why not?

```
int year = 2019;
int yearNext = 2020;
int & ref = year;
ref = yearNext;
```

A.2 - 5 pts - What is a **dangling/stale pointer**? And please give a code example.

Does **`delete`** delete a pointer? [Yes] [No]
Please explain why. Then please give instructions how to properly deallocate an object allocated on Free-Store. Please use a code example to demonstrate.

A.3 - 10 Points

```
#include <iostream>
#include <string>

using namespace std;

string type = "Credit";

class credit_card {
public:
   credit_card() = default;

   explicit credit_card(const double& balance, string com = "Disney") :
      com_(move(com)), balance_(balance) {}

   void display_info() const {
      cout << credit_card::type_ << " [" << this->com_ << "]: " << this->balance_ << endl;
   }

   void set_com(const string& com) {
      this->com_ = com;
   }

private:
   static string type_;
   string com_{ "N/A" };
   double balance_{ 0 };
};

string credit_card::type_ = type;

credit_card& update_credit_card(const double& balance) {
   credit_card cc1{ balance };
   static credit_card* cc2 = new credit_card{ 100 };
   cc1.set_com("Tesla");
   *cc2 = cc1;
   return *cc2;
}

int main()
{
   credit_card cc3 = update_credit_card(300);
   cc3.display_info();

   credit_card cc4 = credit_card{ cc3 };
   cc4.set_com("Zoom");
   cc4.display_info();

   credit_card* cc5 = new credit_card{ update_credit_card(500) };
   cc5->display_info();

   cc3.set_com("Google");
   cc4.display_info();

   return 0;
}
```

- ~~Memory Area 1: Environment~~ *(not tested in this exam)*
- Memory Area **2**: Runtime Stack
- Memory Area **3**: Free-store
- Memory Area **4A**: Uninitialized Data (global, static…)
- Memory Area **4B**: Initialized Data (global, static…)
- ~~Memory Area 5: Binary Program~~ *(not tested in this exam)*

For each element listed below, please answer:

a. In which **memory area** is this element stored? Why?

b. The **lifetime**, begin & end, of this element? Why?

type
Which area: [1] [2] [3] [4a] [4b] [5]
Why is that area?

What is its lifetime and why?

cc2
Which area: [1] [2] [3] [4a] [4b] [5]
Why is that area?

What is its lifetime and why?

cc3, the object
Which area: [1] [2] [3] [4a] [4b] [5]
Why is that area?

What is its lifetime and why?

cc4, the object
Which area: [1] [2] [3] [4a] [4b] [5]
Why is that area?

What is its lifetime and why?

c. What is the **output** of the program?

A.4 - 15 pts – Please **explain each** parameter passing method, **when to use** it, and **code a function** prototype example.

Pass-by-lvalue-reference

Pass-by-const-rvalue-reference

A.5 - 15 pts

int cs = 340;
int *pointer = &cs;

Please code a constant pointer which points to **pointer**

Please code an **lvalue** reference to reference ***pointer**

Please code an **rvalue** reference to reference ***pointer**

Please use what you created, if legal, to change the value of **cs** to 413. **Provide your code** or **reasoning**:

A.6 - 20 Points

```
...
static int x = 1;
int y = x * 2;

void t1() {
	y++;
	cout << "x: " << x << " | y: " << y << endl;
	y += 1;
	x -= -1;
}

void t2() {
	int* x = &y;
	cout << "x: " << x << " | y: " << y << endl;
}

void t3() {
	int y = x;
	static int x = 2;
	cout << "x: " << x + 1 << " | y: " << y + x << endl;
	x += y;
}

void t4() {
	int y = x + 1;
	int& z = y;
	z += -1;
	cout << "x: " << x + z << " | y: " << y << endl;
}

int main() {
	vector<int> vec1{ 1, 3, 5, 7, 9 };
	vector<int> vec2{ 2, 4, 6, 8, 10 };
	vec1.swap(vec2);
	int * ptr = &vec1[1];
	y = *(ptr + 2);

	t1();
	t2();
	t3();
	t3();
	t4();
	return 0;
}
```

This program **outputs** 5 lines. What are they?

1.

2.

3.

4.

5.

A.7 - 20 Points

```
int x = 1, y = -1;

void swapplus1(int n1, int n2) {
	int temp = n1 + 1;
	n1 = n2 - 1;
	n2 = temp;
	x = x + n1;
}
void swapplus2(int& n1, int& n2) {
	int temp = n1 + 1;
	n1 = n2 - 1;
	n2 = temp;
}
void swapplus3(const int& n1, const int& n2) {
	int n1val, n2val, temp = n1 + 1;
	n1val = n2 - 1;
	n2val = temp;
	y -= n2;
}
void swapplus4(int* p1, int* p2) {
	int temp = *p1 + 1;
	*p1 = *p2 + 1;
	*p2 = temp;
	x = *p1 + y;
}
void swapplus5(int* &p1, int* &p2) {
	int* temp = p1 + 1;
	p1 = p2 - 1;
	p2 = temp;
}
void print(const int& x, const int& y) {
	cout << "\n x: " << x << " |y: " << y;
}

int main() {
	int arr[]{ 2, 4, 6, 8, 10, 12, 14 };
	y = arr[3] / size(arr) ;

	swapplus1(x, y);	print(x, y);
	swapplus2(x, y);	print(x, y);
	swapplus3(x, y);	print(x, y);
	swapplus4(&x, &y);	print(x, y);
	int *px = &x, *py = &y;
	(*px)--;
	(*py) -= -7;
	swapplus5(px, py);	print(x, y);
	return 0;
}
```

This program **outputs** 5 lines. What are they?

1.

2.

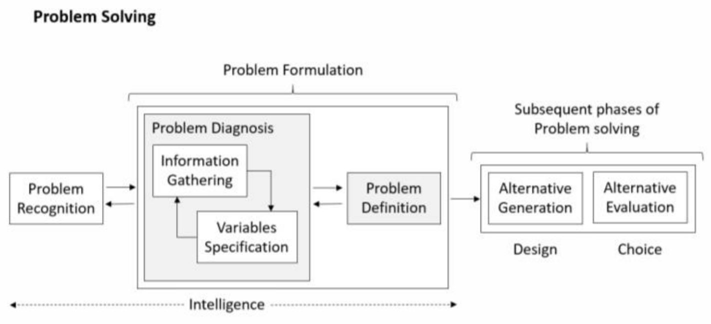
3.

4.

5.

PART B – 10 Points

B.1 - 5 pts



What did the IDEO team do during the Problem Formulation phase? Did you use the problem-solving steps in ASMT 2 and ASMT 3? How or why not?

B.2 - 5 pts

IDEO: In the future, if you will be a CEO of a software company, will you fire people who are better than you are? How will you feel sitting in a meeting with these people? What are the risks if any?