|  |  |  |
| --- | --- | --- |
| Performance Tracker | | |
| Asmt | Grade | Your Grade |
| Zoom | 05 |  |
| 01 | 20 |  |
| 02 | 55 |  |
| 03 | 55 |  |
| Midterm exam | 50 |  |
| Total | 185 |  |
| 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. | | |

Midterm Exam Instructions

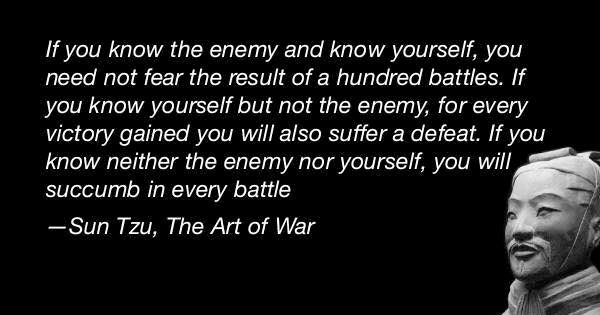
1. Midterm Exam: **50 points w/ 0 E.C. points**
2. Due Date & Time: **07-12-2020 at 6:00 PM**

What to submit

1. Take-home Exam Report (instructions provided below)

How to Submit and the Rules to Follow

* Submit via iLearn, the Assignment 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 to know what we do not know.

Unfortunately, the campus closure does not allow us to host Midterm exam in person. However, we do not have to give up the goal of the exam which is to have an opportunity to evaluate our knowledge of the topics which we covered.

Let us self-evaluate. Then we will go through the questions and the answers together.

We should be able to find an answer to every exam question from the packages, from the in-class discussions, from our assignments, and from the other course materials. It is important that we take this exam no less seriously than how we take an actual exam. We need to know what we do not know. Learn that again to gain a strong foundation for the second half of the semester. Thank you.

PART A – Take the Exam, **20 points**

* Please allocate 115 quiet minutes to take the exam on page 2 to page 9 as seriously as how you take an actual exam.
* Stop right at minute 116 and grade your answers yourself. Please:

1. Record the date and time when you started and stopped taking the exam.
2. (*Do part B then get back to record*) Your score for each question and your total score out of 100.
3. For each answer, list all the mistakes then explain why, you think, you made the mistakes. If you did not make any mistakes, please document how you could have done better.
4. Include your graded exam in this part.
5. **Problem Solving**: Reflect if you managed the exam time efficiently and if you strategized your test-taking successfully.

PART B – Correct Your Answers, **20 points**

* Please review the related course materials and please write your own code to find a correct answer for each question.
* Please document the correct answers, document how you found the correct answers, and document where you found the materials which support the answers.

PART C – Retake the Exam, **10 points**

* Please do all the steps in part A again.
* Repeat part A to part C again if necessary.
* Please think if the same topics will be tested again in Final exam, what questions we may get.

*It is a good idea to do every step of this assignment thoroughly. We are creating a set of materials which we will use to review for Final exams. 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: Full Name in Capital Letters | SFSU ID

CSC 220.01, due 07-12-2020 at 6:00 PM

KAYVAUN KHOSHKHOU

920357344

1. Midterm Exam (1 exam, 0 dropped): 100 points \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. To prepare for this exam, please review all the related materials including WEEK 01-06 packages, slides, mock-up exam(s), reading assignments, in-class practices, sample programs posted in the File Manager, and assignments.
3. You do not need to print this exam. No paper. No handwriting. No scanning. 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.
4. All the rules of an actual exam apply to this exam such as: closed books, closed notes, and no communication with anyone except the course instructor. The course instructor will be available on Zoom during the exam time: zoom.ducta.net
5. Please ask all your questions, if any, during the review sessions. Thank you.

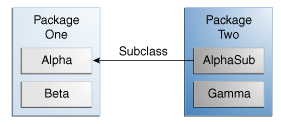
Honor Code:

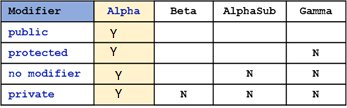
- Please follow the CS Department’s policies: https://cs.sfsu.edu/student-policies  
- Please follow the course’s policies: http://csc220.ducta.net/00-README-StudentConduct\_AcademicHonesty.pdf

Part A – 40 Points

A.1 - 5 Points

Please complete the “where the members of the Alpha class are visible” table by filling in Y or N.

****

****

If a member of the class Beta is protected, is this member visible to Gamma? If yes, please explain why. If not, please explain what we should do make it possible.

Student Answer:  
If a member of class Beta is protected then the member is not visible to Gamma. In order to make it visible to Gamma we need to make its access modifier public. This will allow other classes to see the member.

A.2 - 10 Points

- What are the 2 ways to use the keyword “super” in a class? - Give code examples to demonstrate your answers.

Student Answer:

“super” can be used to invoke parent class methods, as well as to refer parent class instance variables.

We can use “super” in a case like this to refer parent class instance variables:  
  
class Student{  
String favoriteFood = “Pizza”;

}

class Professor extends Student{

String favoriteFood = “Pasta”;

void craving(){

// This will print out “Pasta” since the current class idenitifies the variable “favoriteFood” as such.

System.out.println(favoriteFood);

// With “super” this will print out “Pasta” because it is accessing the Student class to print the data for favoriteFood from that class

System.out.println(super.favoriteFood);

}}

class test{

public static void main(String args[]){

Professor p = new Professor();

p.craving();

}}  
  
OUTPUT:   
Pasta  
Pizza

We can also use “super” in a case like this to invoke parent class methods:

class Student{

void lunch(){

System.out.println(“Time for Pizza!”);

}}

class Professor extends Student{

void lunch(){

System.out.println(“Pasta time!”);

}

void snack(){

System.out.println(“Maybe I’ll just have an apple instead…”);

}

void print(){

super.lunch();

}}

class test{

public static void main (String args[]){

Professor p = new Professor();

p.print();

}}  
  
OUTPUT:  
Time for Pizza!  
Maybe I’ll just have an apple instead…

A.3 - 10 Points

When using a linked list for stack implementation, we use the first Node as the top entry of stack. Explain, in this linked list implementation, how we **add** and how we **remove** an entry from a stack. *Use linked nodes diagrams to save your time.*

**add**

**remove**

Explain one major difference between the behaviors of the LinkedBag and the Stack which we implemented.

A.4 - 5 Points

What are stored in each activation record and Why? Which method is pushed in the Program Stack first? Which method is popped out of the Program Stack last?

A.5 - 10 Points - Part A of Assignment 02:

What is the ouput (what are in the Bag) when the 2 below lines are executed? **Please show the steps.**

String[] items = {"Z", "Y", "Y", "X", "S", "C", "A", "E", "M"};

testAdd(aBag, items);

What is the output (what are in the Bag) when the 3 below lines are executed? **Please show the steps.**

String[] testString = { "X", "Y", "Z" };

aBag.removeAllOccurences(testString);

displayBag(aBag);

Part B – 60 Points

B.1 - 5 Points

Which of the statement(s) are erroneous and why?

**MidtermExam midtermExam = new Exam(); // A**

**SFSUStaff person = new Person(); // B**

**StackInterface<String> s = new ArrayStack<>(); // C**

B.2 - 5 Points

What is the output if any?

**abstract class Person {**

**private final String name;**

**protected Person(String name) {**

**this.name = name;**

**}**

**public String getName() {**

**return this.name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**}**

**class Student extends Person {**

**public Student(String name) {**

**super(name);**

**}**

**public static void main(String[] args) {**

**Student stu = new Student("Mickey");**

**System.out.println(stu.getName());**

**stu.setName("Super Mouse");**

**System.out.println(stu.getName());**

**}**

**}**

*Attention to the keyword “****final****”*

B.3 - 10 Points

Anything wrong with the code? If yes, how to fix?

**class CSC220 {**

**private CSC220(){}**

**private CSC220(int x){}**

**private CSC220(int x, int y){}**

**}**

And is it possible to create a sub class to this class? Why?

B.4 - 10 Points

What is the output if any?

**public static void main(String[] args) {**

**int x = 12;**

**System.out.println(sumOf(x) - 42);**

**}**

**public static int sumOf(int n) {**

**int sum = 15;**

**if (n == 0) {**

**System.out.println("Base case: n is " + n);**

**sum += 5 + n % 2;**

**} else {**

**sum = sumOf(n - 3) + n;**

**}**

**System.out.println(sum);**

**return sum;**

**}**

B.5 - 10 Points

This program outputs 10 lines. What are they?

**Stack<String> resume = new Stack<>();**

**resume.push("JavaScript");**

**System.out.println("Is empty: \t" + resume.isEmpty());**

**resume.push("Scala");**

**resume.push("C++");**

**resume.push("Dart");**

**resume.push("Go");**

**resume.pop();**

**System.out.println("Stack : \t" + resume);**

**resume.push("Python");**

**System.out.println("search() : \t" + resume.search("Scala"));**

**System.out.println("pop() : \t" + resume.pop());**

**System.out.println("pop() : \t" + resume.pop());**

**System.out.println("search() : \t" + resume.search("Dart"));**

**System.out.println("After pop() : \t" + resume);**

**System.out.println("pop() : \t" + resume.pop());**

**System.out.println("Is empty : \t" + resume.isEmpty());**

**System.out.println("Stack: \t" + resume);**

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

B.6 - 10 Points

Implement *findTheThird* method in linked list that searches the bag for a given *entry.*  If found,

* removes the first occurrence
* leave the second occurrence intact
* then replace third occurrence with the string “Found3rd”
* remove the rest of the occurrences

Return *false* if no replacement happened. Otherwise, *true*.

**public boolean findTheThird (T entry)**

Note: You may assume that *firstNode* is a private data in list

which references to first node.

B.7 - 10 Points

Assume that you have the following Bag object, myBag, with n String data:

BagInterface <String> myBag = new ArrayBag< >();

Write Java statements that create a newBag object which

contains non-duplicate data in myBag and marks the duplicate data.

Example:

if myBag contains data:

**"A", "A", "A", "B", "B", "C", "D", " "**

newBag object should contain:

**“A”, “DUP.1.A”, “DUP.2.A”, “B”, “DUP.3.B”, “C”, “D”, “ “**

Hint: You can use the Bag’s methods:

int getCurrentSize (); boolean isFull (); boolean isEmpty (); boolean add (T newEntry); T remove (); boolean remove (T anEntry); void clear (); int getFrequencyOf (T anEntry);

boolean contains (T anEntry); T [] toArray ();