MIDTERM EXAM INSTRUCTIONS

1. Midterm Exam: **25 points w/ 0 E.C. points**
2. Due Date & Time: **10-19-2024 at 11:55 PM**

WHAT TO SUBMIT
1. Midterm Exam Report, 1 PDF
   Example: DucTa-Exam-Midterm-Report.pdf

HOW TO SUBMIT AND THE RULES TO FOLLOW
- The Guidelines for All Assignments
- The Course Policy on Student Conduct and Academic Honesty
- The assignment instructions and rubric for this assignment
- The additional instructions are provided in class and on Canvas.
- Submit via Canvas, the Assignment Submission section.

PERFORMANCE TRACKER		
ASMT	**GRADE**	**YOUR GRADE**
CANVAS	15	
DL-APPROVAL-01	COMPLETE	
01	25	
DL-APPROVAL-02	COMPLETE	
02	30	
03-PREPARATION	25	
03	50	
ADVISING	25	
04	50	
05	40	
MIDTERM EXAM	25	
TOTAL	**285**	

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

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 we take an actual exam in class. Please,
1. Follow all the rules and guidelines listed at the top of page 1 and page 2
2. Read each question carefully before answering

After the exam, we will go over the answers together in class.

STEP A – Take the Exam, **10 points**

1. Allocate 50 quiet minutes to take the exam from 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.

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. **Confirm you have done this step B.1 carefully**.

```
EXAM, General Layout

                Full Name | SFSU ID

- Exam, Start, Date and Time:
- Exam, Stop, Date and Time:
- Exam Score for the Original Answers:

- Each question
      - Step A: Answer
      - Step B: Correct Step A's answer
      - Step C: More practice
      - Other notes

- Next question
```

2. At the end of each of your original answers, type in *italic* text and
   - Give your original 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 that 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 that support the answer.
   - Outline how you could have done better.
   - **Show all your work in detail**. Use the provided Microsoft Word exam file/template. Space was provided for answers.

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

1. **Problem-Solving**: Reflect on whether you managed the exam time efficiently and strategized your test-taking successfully.
2. Repeat steps A to C if necessary. Please keep appending new content as directed in Step B.2.
3. Think, if the same topics will be tested again in our final exam and your job interviews, 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 that 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
   Section 28. Due 10-19-2024 at 11:59 PM

Full Name in CAPITAL LETTERS | SFSU ID

MIGUEL ANTONIO LOGARTA	923062683

2. Midterm Exam (1 exam, 0 dropped): 25 Points

3. To prepare for this exam, please review all the related materials including the packages, slides, mock exam, reading assignments, in-class practices, sample programs posted in the File Manager, and assignments.

4. Do not 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. Again, use the provided exam file, not the assignment template. This is an exam, not an assignment.

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 via email during the exam time.

6. Again, you cannot use any other materials or tools but only the provided exam which will be in Microsoft Word format.

7. Please ask all your questions, if any, during the review sessions. Thank you.

HONOR CODE:

- Please strictly follow:

  - The Course Policy on Student Conduct and Academic Honesty
  - The instructions which are given in class, on Canvas, and during the exam

---

- Exam, **Start** at Date and Time: 10-19-2024 at 01:35 PM
- Exam, **Stop** at Date and Time: 10-19-2024 at 02:15 PM
- Exam **Total Score** for the Original Answers: ## / 100

* The final score for this take-home exam is the sum of the points gathered from Step A, Step B, and Step C (Page 1). *The points offered to each question (such as Part A.1 – 5 pts) are only references and are to help us prepare for the final exam.*

PART A – 40 Points

A.1 - 8 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

How is a **programmer** different from a **coder**? How is **EQ** different from **IQ**?

A coder is only concerned about making sure that the program that they make is up to spec. They only write about code and doesn't concern themselves about the other aspects of the program. What sets the differences between programmers and coders are that programmers also have to care about the structure and scalability of the overall program. Programmers have to deal with the client's needs, wants, and concerns. Overall, the programmer has more responsibility than the coder.

IQ is logical intelligence while EQ is emotional intelligence. High IQ intelligence means that you can efficiently solve problems and tasks faster than other people, while high EQ means that you know how to solve conflicts with others, identify emotions, and empathize with people better than everybody else. High EQ is needed to communicate effectively with other people which is important for environments where you have to constantly face clients.

A.2 - 8 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Write a complete and working method that is **not** a **void** method. Then name and explain each component of it. Does a **void** method return anything?

The function below adds two numbers a and b and returns the sum.

```
public static int add(int a, int b) {
     return a  + b;
}
```

The function follows this structure:
```
<access_modifier> <non_access_modifier> <return_type> <function_name>(<parameter_list>) {
     <function_body>
}
```

access_modifier *determines whether the method can be accessed outside of the class or not. Public means we can access it outside of the class*
non_access_modifier *determines the behavior of the function. Static means that we don't have to instantiate the class to access this method.*
return_type *is the data type that the function returns. Add() returns an integer.*
function_name *is the name of the function*
parameter_list *is the inputs of the method. We have two parameters a and b which are both of integer data type.*
method_body *is the code that gets executed when we call the method. Add() adds a and b together and returns the result.*

A **void** *method doesn't return anything. When we call a void function, it executes code, but it doesn't return any data.*

A.3 - 8 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Please write code to declare and initialize a **1D** array to contain more than 37 items. Then write a **for** loop to display the content of the array. The control variable(s) must be initialized to a value greater than 17.

```
int N = 37;
int START = 17;

int[] arr = new int[N];

for (int I > START; I < N; I++) {
     arr[I] = I;
}

for (int I > START; I < N; I++) {
     system.out.println(arr[I]);
}
```

A.4 - 8 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Please write code to declare and initialize a **2D** array to contain at least 7 rows and 71 columns in each row. Then write a **for** loop to display the content of the array.

```
Int N = 7;
int M = 71;
int[][] arr = new int[7][71];
int value = 0;

for (int I = 0; I < N; I++) {
    for (int j = 0; j < M; j++ {
        arr[I][j] = value;
        value++;
    }
}

for (int I = 0; I < N; I++) {
    for (int j = 0; j < M; j++ {
        system.out.println(arr[I][j]);
    }
}
```

A.5 - 8 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Please list and explain the general components of a Java class that represents **student workers**.

```
Public class StudentWorkers { // class name
    double grades;    // class fields
    double payRate;

    StudentWorkers(double grades, double payRate) {    // constructor
        ...
    }

    public void doHomework() {    // class method
        ...
    }

    public double getPayRate() {    // class method (getter)
        ...
    }
}
```

The java class that represents student workers is represented in a class called StudentWorkers.

The general components of a java class include the class name, contructor, class fields, and class methods

class name: Class name is the name of the class.
Class fields: Class fields are the variables that are inside of the class. Every object that is created from this class have these variables with their own values separate from other objects.
Constructor: This is the method that gets called when we create an object of this class. It contains code that helps sets up the object when it is created. Usually, the code would try to instantiate the class fields to some value.
Class methods: These are the functions inside of the class. When you call the class method it usually modifies some variables within the class.

<div align="center">

PART B – 60 Points

</div>

B.1 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Please code a complete **while** loop with a meaningful body then convert it into a **do-while** loop and a **for** loop. These loops must iterate an array by starting at the **3nd** item from the **left** and skipping the **7th** item if it exists.

while loop

```java
int[] values = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
int counter = 3;

while (counter < values.length) {
    if (counter != 7) {
        system.out.println(values[counter]);
    }
    counter++;
}
```

do-while loop

```java
int[] values = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
int counter = 3;

do {      // Will always print at least 1 element unless counter == 7.
    if (counter != 7) {
        system.out.println(values[counter]);
    }
    counter++;
} while (counter < values.length);
```

for loop

```java
int[] values = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};

for (int  I = 3; I < values.length; i++) {
    if (i != 7) {
        system.out.println(values[i]);
    }
}
```

B.2 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Please code a complete statement containing a **switch expression** then convert it into an **if-else** statement. The **switch expression** must have at least 5 cases (including the **default** case).

switch expression in a complete statement

```
String grade;

...

switch(grade) {
    case "A":
        System.out.println("You did amazing!");
        break;
    case "B":
        System.out.println("You did good!");
        break;
    case "C":
        System.out.println("You did okay...");
        break;
    case "D":
        System.out.println("You need to do better...");
    default:
        System.out.println("You failed!");
}
```

if-else statement

```
String grade = "";

...

if (grade == "A") {
    System.out.println("You did amazing!");
} else if (grade == "B") {
    System.out.println("You did good!");
} else if (grade == "C") {
    System.out.println("You did okay...");
} else if (grade == "D") {
    System.out.println("You need to do better...");
} else {
    System.out.println("You failed!");
}
```

B.3 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Please code a complete Java program: **EnrollmentOrNoEnrollment**
- Your program prompts users to answer if they have the time to succeed if they enroll in the course.
- Then the program prints a recommendation based on the user's previous input.
- It is OK to assume that users will enter a valid response.
- This program must have at least 4 methods. (1 of them is the main method.)
- A sample run of the program (think the Agree/Disagree buttons to Terms of Service):

If enrolled in this course, do you have the time to succeed? Yes
For every hour in class, allocate at least 5 hours to study outside of class.

Note: "Yes" is one valid choice. You can choose a different choice for your code (and sample run).

```
Public class EnrollmentOrNoEnrollment {
    int askHoursLeft() {
        Scanner sc = new Scanner(system.in);
        system.out.print("How many hours left do you have free in the week? ");
        return sc.nextInt();
    }

    int askHours() {
        Scanner sc = new Scanner(system.in);
        system.out.print("How many hours are in this class? ");
        return sc.nextInt();
    }

    String askTime() {
        Scanner sc = new Scanner(system.in);
        system.out.print("If enrolled in this course, do you have the time to succeed? ");
        return sc.nextLine().trim().toLowerCase();
    }

    int getRecommendation() {
        String userResponse = askTime();
        if (userResponse == "no") {
            system.out.println("If you feel you can't succeed, then I recommend you don't enroll.");
        } else {
            int classHours;
            int weekHours;
            system.out.println("For every hour in class, allocate at least 5 hours to study outside of
class.");
            classHours = askHours();
            weekHours = askHoursLeft();

            if (weekHours – (classHours*5) < 0) {
                system.out.println("This class will be difficult since you don't have enough time to
study! Make sure you are extra prepared before you enroll.");
            } else {
                system.out.println("This class is perfect. You should enroll!");
            }
        }
    }

    public static void main(String args[]) {
        getRecomendation();
    }
}
```

Sample run:

```
If enrolled in this course, do you have the time to succeed? Yes
For every hour in class, allocate at least 5 hours to study outside of class.
How many hours are in this class? 2
How many hours left do you have free in the week? 15
This class is perfect. You should enroll!

If enrolled in this course, do you have the time to succeed? Yes
For every hour in class, allocate at least 5 hours to study outside of class.
How many hours are in this class? 2
How many hours left do you have free in the week? 8
This class will be difficult since you don't have enough time to study! Make sure you are extra
prepared before you enroll.

If enrolled in this course, do you have the time to succeed? No
If you feel you can't succeed, then I recommend you don't enroll.
```

B.4 TO B.6 QUESTIONS ARE LINKED TO EACH OTHER

B.4 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Please choose a real-life entity then write a Java class to represent it. The entity must be part of the **J. Paul Leonard Library** at **San Francisco State University**.

What is that entity?

For my entity, I am going to select a Bookshelf entity.

Why do you think it is suitable to be a Java class?

I think it is suitable for a Java class because a bookshelf can contain an inventory of different Books. You can also interact with the bookshelf by filling it up with books, or taking away books to read them.

Please code the class. Your class should have data fields, constructors, and methods.

Public class Bookshelf {
    String[] books;
    int capacity;
    int maxCapacity;

    Bookshelf(String[] newBooks, int newMaxCapacity) {
        if (newMaxCapacity > newBooks.length) {
            this.books = new String[newMaxCapacity];
            this.capacity = newBooks.length;
            for (int I = 0; I < newBooks.length; i++) {
                this.books[i] = newBooks[i];
            }

```
        } else {
            this.books = newBooks;
            this.maxCapacity = newBooks.length;
        }
    }

    public void addBook(String newBook) {
        if (this.capacity == this.maxCapacity) {
            system.out.println("Bookshelf is full!");
        } else {
            for (int I = 0; I < this.books.length; i++) {
                if (this.books[i] == null) {
                    this.books[i] = newBook;
                    break;
                }
            }
            this.capacity = this.capacity + 1;
            system.out.print("%s Added", newBook);
        }
    }

    public void removeBook(String book) {
        boolean bookFound = false;
        for (int I = 0; I < this.books.length; i++) {
            if (this.books[i] == book) {
                this.books[i] = null;
                bookFound = true;
                break;
            }
        }

        if (bookFound) {
            this.capacity = this.capacity - 1;
            system.out.print("%s Removed", book);
        } else {
            system.out.printf("%s Not found", book);
        }
    }

    public void printBooks() {
        for (String book : this.books) {
            system.out.println(book);
        }
    }
}
```

B.5 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Please write code to add another data field to your class to keep track of the number of objects created.

```
Public class Bookshelf {
    public String[] books;
    public int capacity;
```

```java
public int maxCapacity;
public static int numBookshelves = 0;

Bookshelf(String[] newBooks, int newMaxCapacity) {
    Bookshelf.numBookshelves  = Bookshelf.numBookshelves + 1;
    if (newMaxCapacity > newBooks.length) {
        this.books = new String[newMaxCapacity];
        this.capacity = newBooks.length;
        for (int I = 0; I < newBooks.length; i++) {
            this.books[i] = newBooks[i];
        }
    } else {
        this.books = newBooks;
        this.maxCapacity = newBooks.length;
    }
}

public void addBook(String newBook) {
    if (this.capacity == this.maxCapacity) {
        system.out.println("Bookshelf is full!");
    } else {
        for (int I = 0; I < this.books.length; i++) {
            if (this.books[i] == null) {
                this.books[i] = newBook;
                break;
            }
        }
        this.capacity = this.capacity + 1;
        system.out.print("%s added", newBook);
    }
}

public void removeBook(String book) {
    boolean bookFound = false;
    for (int I = 0; I < this.books.length; i++) {
        if (this.books[i] == book) {
            this.books[i] = null;
            bookFound = true;
            break;
        }
    }

    if (bookFound) {
        this.capacity = this.capacity - 1;
        system.out.print("%s Removed", book);
    } else {
        system.out.printf("%s Not found", book);
    }
}

public void printBooks() {
    for (String book : this.books) {
        system.out.println(book);
    }
}
}
```

Please explain in detail how the data field should be used. Provide code to demonstrate and support your explanation.

Since the data field numBookshelves is static, the variable is shared across all the objects. Whenever we create a new Object of the Bookshelf class, numBookshelves increases by 1. We can get the number of total bookshelves that we have by referring to the data field by "Bookshelf.numBookshelves"

Then explain in detail why your code should work properly. Provide code to demonstrate and support your explanation.

With my code, you can instantiate a Bookshelf with a custom capacity with some books. You can add and remove books from a bookshelf. Here's how to use the Bookshelf class:

Bookshelf fictionBooks = new Bookshelf(new String[]{"Harry Potter", "Dune", "Lord of the Rings", "The Witcher", "A Song of Ice and Fire"}, 10); // Creates bookshelf that can hold at most 10 books
fictionBooks.addBook("The Hobbit"); // Adds the hobbit
fictionBooks.printBooks();
fictionBooks.removeBook("Dune"); // removes dune
fictionBooks.printBooks();
System.out.println(Bookshelf.numBookshelves); // 1

The contents of the bookshelf is:
Harry Potter
null
Lord of the Rings
The Witcher
A Song of Ice and Fire
The Hobbit
null
null
null
null

B.6 - 10 Points – *Your answer must be in your own words, be in complete sentences, and provide very specific details to earn credit.*

Write a no-argument constructor, a two-argument constructor, and a four-argument constructor for your class.

```
Bookshelf() {
    Bookshelf.numBookshelves  = Bookshelf.numBookshelves + 1;
    this.books = new String[10];
    this.capacity = 0;
    this.maxCapacity = 10;
}

Bookshelf(String[] newBooks, int newMaxCapacity) {
        Bookshelf.numBookshelves  = Bookshelf.numBookshelves + 1;
        if (newMaxCapacity > newBooks.length) {
            this.books = new String[newMaxCapacity];
            this.capacity = newBooks.length;
            for (int I = 0; I < newBooks.length; i++) {
                this.books[i] = newBooks[i];
            }
        } else {
```

```
            this.books = newBooks;
            this.maxCapacity = newBooks.length;
        }
    }

Bookshelf(String[] newBooks, int newCapacity, int newMaxCapacity, int numBookshelves) {
        Bookshelf.numBookshelves  = numBookshelves;
        if (newMaxCapacity > newCapacity) {
            this.books = new String[newMaxCapacity];
            this.capacity = newCapacity;
            for (int I = 0; I < newBooks.length; i++) {
                this.books[i] = newBooks[i];
            }
        } else {
            this.books = newBooks;
            this.maxCapacity = newMaxCapacity;
        }
    }
```

Code instructions to create 3 objects using all constructors above.

```
Bookshelf emptyBookshelf = new Bookshelf;
Bookshelf fictionBooks = new Bookshelf(new String[]{"Harry Potter", "Dune", "Lord of the Rings", "The Witcher", "A Song of Ice and Fire"}, 10); // Creates bookshelf that can hold at most 10 books
Bookshelf bigBookshelf  = new Bookshelf(new String[]{"Hamlet", "The Great Gatsby"}, 2, 50, 3);
```

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