Midterm 1

Note: You don't have to complete all of the questions! Focus on the topics you need the most help with and navigate using the document tabs on the left-hand side!

Largest Weaknesses From Midterm 1(last updated S26):

1. Math.random()
2. Counting Operations
3. Identifying Compile-time Errors

Pseudocode Counting Operations

S25 - 1.1c: Regular: ~54%, Conflict: ~53%

Section 1: Pseudocode with Conditionals (25 points)

1. **(12 points)** Read the following algorithm that calculates and displays how many minutes of television Janiya will be allowed to watch before bedtime.

```
READ choresDone
READ homeworkDone
READ minutesWatched

SET tvTime TO 10

IF choresDone IS true THEN
        ADD 15 TO tvTime
ENDIF
IF homeworkDone IS true THEN
        ADD 30 TO tvTime
ELSE
        SUBTRACT 5 FROM tvTime
ENDIF
IF minutesWatched > 0 THEN
        SUBTRACT minutesWatched FROM tvTime
ENDIF

DISPLAY tvTime
```

a) **(2 points each)** Determine the outputs for each of the following inputs. The inputs are listed in the order that they're read in the pseudocode algorithm.

1. true, true, 5 → _____
2. true, false, 0 → _____
3. false, true, 10 → _____

b) **(3 points)** What is the maximum number of operations that can be executed?
Answer:

c) **(3 points)** What is the minimum number of operations that can be executed?
Answer:

S25 - 2.1c: Regular: ~44%, Conflict: ~36%

Section 2: Pseudocode with Loops (25 points)

1. **(15 points)** Consider the following pseudocode algorithm that counts how many integers between 1 (inclusive) and n (inclusive) are divisible by 3. Assume n is always greater than 1.

```
READ n
SET count TO 0
SET i TO 1
WHILE (i <= n)
        IF (i % 3 IS 0) THEN
                ADD 1 TO count
        ENDIF
        ADD 1 TO i
ENDWHILE
```

c) **(7 points)** Assume x is the number of times i is divisible by 3. What is the total number of operations executed in terms of n and x?

S24 - 1.1c: Regular: ~59%, Conflict: ~51%

The four variables read are all boolean variables. Find the minimum amount of operations.

```
READ bedMade
READ laundryWashed
READ clothesFolded
READ dishesWashed

SET allowance TO 0

IF bedMade IS true THEN
      ADD 2 TO allowance
ENDIF

IF laundryWashed IS true THEN
      IF clothesFolded IS true THEN
            ADD 5 TO allowance
            ELSE
            ADD 2 TO allowance
      ENDIF
ENDIF

IF dishesWashed IS true THEN
      ADD 3 TO allowance
ENDIF

DISPLAY allowance
```

Pseudocode Counting Operations Answers

Answer S25 - 1.1c: 9
Answer S25 - 2.1c: 3n + 4 + x
Answer S24 - 1.1c: 9

Java Basics

S25 - 3.2d: Regular:~50% correct

2. **(9 points)**
Consider the following java program that reads in 2 **integer** inputs and computes their average.

```
public class Average{
    Run | Debug
    public static void main(String[] args){
        int a = Integer.parseInt(args[0]);
        int b = Integer.parseInt(args[1]);
        int average = (a + b)/2;
        System.out.println(average);
    }
}
```

d) **(3 points)** How could the program be fixed so that the correct average would be printed for any two integer values input by the user? Write the modified line(s) of Java code.

S25 - 3.2a: Conflict: ~25%
S25 - 3.2b: Conflict: ~27%

2. **(8 points total)**

a) **(3 points)** In Java, what is a **.class** file, and how is it created?

A programmer wishes to display the quotient of two integer values input by the user. Her code segment below does not work at intended.

```
37      int num1 = Integer.parseInt(args[0]);
38      int num2 = Integer.parseInt(args[1]);
39      double result = num1/num2;
40      System.out.println(result);
```

b) **(2 points)** What would the output of this code segment be if the user entered the values 5 2 in that order?

F24 - 4.1: Regular: ~41%, Conflict: ~45%

Section 4: Coding Conditionals (40 points)

1. **(9 points – 3 points each)** Consider the program below. Assume the program has been compiled and that the following terminal commands are used to run the program. Predict the output for each command.

```
int val1 = Integer.parseInt(args[0]);
int val2 = Integer.parseInt(args[1]);
if (val1 < 0 || val2 < 0){
    System.out.println("negative pair");
}
if (val1 > 0 && val2 > 0){
    System.out.println("positive pair");
}
else{
    System.out.println("neither");
}
```

a) java QA4 5 10 _____

b) java QA4 -5 -10 _____

c) java QA4 5 -10 _____

S24 - 2.1b: ~60% on regular, ~51% on conflict

1. **(12 points)** The following pseudocode algorithm counts how many integers (between 1 and n inclusive) are divisible by both 3 and 4.

```
READ n
SET i TO 1
SET counter TO 0
WHILE i <= n
      IF ( i % 3 IS 0 ) AND ( i % 4 IS 0 ) THEN
            ADD 1 TO counter
      ENDIF
      ADD 1 TO i
ENDWHILE
```

b) **(6 points)** Let's say *x* is the number of integers divisible by both 3 and 4. What is the total number of operations executed in terms of *n* and *x*? **HINT:** `ADD 1 TO counter` is only executed if the number is divisible by both 3 and 4. Answer: _____

Java Basics Answers

Answer S25 - 3.2d:

```
double average = (a + b)/2.0; OR
double average = (double) (a + b)/2
```

```
//no partial credit
```

Answer S25 - 3.2:
a) It is the bytecode representation of the Java program/file only the computer can read(1 pt)
generated with the javac command(2 pt)
b) 2.0 → they must have 2.0 and not 2

Answer F24 - 4.1:
a)
positive pair
b)
negative pair
neither
c)
negative pair
neither

Answer S24 - 2.1b: 4n + 4 + x

Math.random()

S25 - 3.3: Regular: ~49%, Conflict: 65%

3. **(5 points)** One person of 20 people in a room will be given a free 2025 season pass to Rutgers Football games. Each of the 20 people is given a card with an integer value between 1 and 20. No number is duplicated and each of the numbers from 1 to 20 is assigned to a person.

The following java code segment is designed to select and print an integer value between 1 and 20 inclusive [1,20] that will determine the lucky person.

```
1   public class Rands{
        Run | Debug
2       public static void main(String[] args){
3           double rand = Math.random() * 20;
4           System.out.println(rand);
5       }
6   }
```

This code does not work as intended. Instead, it prints a double value between 0 and 20 – not including 20: [0,20). **Rewrite line 3** so that the intended result is printed: an **integer** between 1 and 20 **inclusive** [1,20].

S24 - 3.4: Regular: ~60%, Conflict: ~66%

4. **(9 points)** Write a snippet of code in Java that produces a random number between 1 and 10 and determines if it is even or odd. Your program will display *true* if an even number is generated, and *false* otherwise.

- You MAY NOT use if statements/loops. Answers including these will not receive credit.
- You DO NOT need to include the headers (public class/public static void main).
- **HINT:** `Math.random()` produces a random double number between 0.0 (inclusive) and 1.0 (exclusive).

S24 - 5.2: Conflict Exam: ~23%

2. **(8 points)** The program Guess simulates a two person game of guessing numbers between 1 and 100 inclusive. One person thinks of a number, the other guesses it.

The program takes a positive integer command line argument as input, checks if it is within the 1-100 range, and then randomly generates a number between 0-100 inclusive. If the input is the same as the randomly generated number, then the message "You guessed it right!" is displayed. If the numbers do not match then "No luck, try again." is displayed.

The program contains a compiler error. It means that when compiled with the command java Guess.java an error will be displayed.

```
1   public class Guess{
        Run | Debug
2       public static void main(String[] args){
3
4           int guessedNumber = Integer.parseInt(args[0]);
5
6           //guessedNumber must be in [0,100]
7           if ( guessedNumber < 0 || guessedNumber > 100 ){
8               System.out.println("Guessed value out of range");
9           } else {
10
11              //Generate a number between 0.0 and 100.0 then cast to
12              //an integer variable
13              //value will contain values from 0 to 100 inclusive
14              int value = (Math.random() * 101);
15
16              if ( guessedNumber == value ){
17                  System.out.println("You guessed it right!");
18              } else {
19                  System.out.println("No luck, try again.");
20              }
21          }
22      }
23  }
```

What is the error and which line(s) of code need to be changed for it to work as intended? You may use the line number to let us know which line needs to be changed and how to change it.

Math.random() Answers

Answer S25 - 3.3:

```
int rand = (int) (Math.random() * 20) + 1
```

Answer S24 - 3.4:

```
int rand = (int) (Math.random() * 10) + 1; //4 points for correct random #, -1 for forgetting + 1
 boolean bool = (rand % 2 == 0); //3 points for boolean
 System.out.println(bool); //2 points displaying
```

Answer S24 - 5.2 Conflict:

Line 14 needs to be updated, cast Math.random() to int

int value = (int) (Math.random() * 101);

(no partial credit)

Modulus

S25 - 3.4: Regular: ~59%

4. (14 points)

Complete the Java program below that is intended to calculate and display the equivalent number of hours, minutes, and seconds for a given input of seconds. There are 60 seconds in one minute and 60 minutes in one hour. The result will be displayed in a descriptive print statement.

Example runs:

```
$ java Seconds 3800
1 hours 3 minutes 20 seconds
```

```
$ java Seconds 2025
0 hours 33 minutes 45 seconds
```

```
$ java Seconds 8673
2 hours 24 minutes 33 seconds
```

```
public class Seconds{
    Run | Debug
    public static void main(String[] args){
        int totalSec = Integer.parseInt(args[0]);
        //WRITE YOUR CODE HERE
```

S25 - 3.4: Conflict: ~55%

4. (14 points) Complete the Java program **Measures** below that is intended to calculate and display the equivalent number of yards, feet, and inches for a given input of inches. There are 12 inches in one foot and 3 feet in 1 yard. The result is displayed in a descriptive print statement.

Example runs:

```
$ java Measures 65
1 yards 2 feet 5 inches
```

```
$ java Measures 2025
56 yards 0 feet 9 inches
```

```
$ java Measures 35
0 yards 2 feet 11 inches
```

```
public class Measures{
    Run | Debug
    public static void main(String[] args){
        int totalInches = Integer.parseInt(args[0]);
```

Modulus Answers

Answer S25 Regular- 3.4:

```
int hours = totalSec / 3600;
totalSec -= hours * 3600;
int minutes = totalSec / 60;
totalSec -= minutes * 60;
System.out.println(hours + " hours "
                 + minutes + " minutes "
                 + totalSec + " seconds");
```

```
int hours = totalSec / 3600;
totalSec = totalSec % 3600;
int minutes = totalSec / 60;
totalSec = totalSec % 60;
System.out.println(hours + " hours "
                 + minutes + " minutes "
                 + totalSec + " seconds");
```

```
int minutes= totalSecs / 60;
int secs = totalSecs % 60;
int hours = minutes / 60;
minutes = minutes % 60;
System.out.println(hours + " hours "
                 + minutes + " minutes "
                 + secs + " seconds");
```

Answer S25 Conflict - 3.4:

```
int yards = totalInches/36;
totalInches = totalInches % 36;
```

```
int feet = totalInches / 12;
int inches = totalInches % 12;
System.out.println(yards + " yards "
                   + feet + " feet "
                   + inches + " inches");


int feet2 = totalInches2 / 12;
int inches2 = totalInches % 12;
int yards2 = feet2 / 3;
feet2 = feet2 % 3;
System.out.println(yards2 + " yards "
                   + feet2 + " feet "
                   + inches2 + " inches");
```

4 pt for descriptive print – use of concatenation

Compile Time Error

F24 - 5.1a: Regular: ~43%, Conflict: 47%

1. **(20 points)** The following simple Java program is intended to print even numbers from 2 to 100 using a for loop, but it does not compile.

```
for (i = 0; i < 100; i++);
{
    if (i % 2 != 0){
        System.out.println(i);
    }
}
```

a) **(4 points)** What will cause a compile-time error if this program is compiled?

S24 - 5.2: Regular: ~55%

2. **(8 points)** The program Guess simulates a two person game of guessing numbers between 1 and 100 inclusive. One person thinks of a number, the other guesses it.

The program takes a positive integer command line argument as input, checks if it is within the 1-100 range, and then randomly generates a number between 0-100 inclusive. If the input is the same as the randomly generated number, then the message "You guessed it right!" is displayed. If the numbers do not match then "No luck, try again." is displayed.

The program contains a compiler error. It means that when compiled with the command `javac Guess.java` an error will be displayed.

```
1   public class Guess{
        Run | Debug
2       public static void main(String[] args){
3
4           int guessedNumber = Integer.parseInt(args[0]);
5
6           //guessedNumber must be in [0,100]
7           if ( guessedNumber < 0 || guessedNumber > 100 ){
8               System.out.println("Guessed value out of range");
9           } else {
10
11              //Generate a number between 0.0 and 100.0 then cast to
12              //an integer variable
13              //value will contain values from 0 to 100 inclusive
14              int value = (int) (Math.random() * 101);
15
16              if ( guessedNumber = value ){
17                  System.out.println("You guessed it right!");
18              } else {
19                  System.out.println("No luck, try again.");
20              }
21          }
22
23      }
24
25
```

What is the error and which line(s) of code need to be changed for it to work as intended? You may use the line number to let us know which line needs to be changed and how to change it.

Compile Time Error Answers

Answer F24 - 5.1a:
the variable 'i' is not initialized

Answer: Line 16 needs to be updated, replace `=` by `==`
(no partial credit)

Midterm 2

Note: You don't have to complete all of the questions! Focus on the topics you need the most help with and navigate using the document tabs on the left-hand side!

Largest Weaknesses for Midterm 2 (last updated in S25):

1. Piping
2. Tracing Code
3. One-Dimensional Arrays

Tracing Java Code

F24 - 2.2b: Regular ~61%, Conflict ~51%

2. **(15 points)** Consider the following code snippet for parts a and b.

```
int i = 1234;
int j = 1;
int k = 0;
while (i > 0){
    k = i % 10;
    if (k % 2 == 0)
        j *= k;
    i = i / 10;
    System.out.println(k + " " + j + " " + i);
}
```

a) **(12 points)** Use the table below to trace the code execution. To do so, enter values printed for i, j, and k at the end of each iteration. You may not need all rows in the table.

Iteration	k	j	i
1			
2			
3			
4			
5			

b) **(3 points)** In one short phrase, describe the final value stored in variable `j` in terms of any number `i`.

F24 - 5.1: Regular ~55%, Conflict ~52%

Section 5: Writing Functions (35 points)

1. **(5 points)** What is printed when the function `foo()` is executed?

```
public static void foo () {
    int k = 2;
    bar(k);
    System.out.println(k);
}

public static void bar (int k) {
    k = k * k;
    System.out.println(k);
}
```

Output:

Tracing Java Code Answers

Answer F24 - 2.2:

Iteration	k	j	i
1	4	4	123
2	3	4	12
3	2	8	1
4	1	8	0
5			

b) **(3 points)** In one short phrase, describe the final value stored in variable `j` in terms of any number `i`.

The program computes the product of all even digits in i.
Alternate: multiplies all even numbers in i. //no partial credit

Answer F24 - 5.1: 4 2

Piping

F24 - 4.3b: Regular ~58%, Conflict ~52%

3. **(5 points)** Consider the Java program, Piping.java, and text file, input.txt, below:

```
public class Piping{
    Run | Debug
    public static void main(String[] args){
        int x = StdIn.readInt();
        int y = StdIn.readInt();
        StdOut.println(y + " 1 " + x);
    }
}
```

input.txt

```
6
1
5
```

b) **(3 points)** What will be displayed when the command `java Piping < input.txt | java Piping` is executed? Circle your final answer.

S24 - 4.2: Regular: ~36%, Conflict: ~31%

2. **(10 points)** What is the output of the program below when executed with the command **java M2 < input.txt**?

```
public class M2{
    Run | Debug
    public static void main(String[] args){

        int    n = StdIn.readInt();
        double x = StdIn.readDouble();
        double z = 0.0;
        int    c = 0;

        while (z < x && c < n) {
            z += StdIn.readDouble();
            c += 1;
        }
        StdOut.println(z);

    }
```

input.txt

```
6
5.0
2.5
1.0
4.0
1.5
3.0
2.5
```

Answer:

Piping Answers

Answer F24 - 4.3b: 1 1 1
Answer: 7.5

Java Basics/Arrays

F24 - 5.2: Regular: ~63%, Conflict: ~47%

Try doing the same but instead of rounding down, have the function round up.

2. (6 points) Write the function `floor` that takes as input a positive double number and returns the greatest integer less than or equal to the number parameter. Assume that the input number is greater than zero. You may NOT use the Math class. For example:

- `floor (2.36)` returns `2`
- `floor (5.0)` returns `5`

```
public static int floor (double number){




}
```

S24 - 3.2b: Regular: ~62%, Conflict: ~64%
S24 - 3.2c: Regular: ~59 %, Conflict: ~56%
S24 - 3.2d: Regular: ~64%, Conflict: ~69%

2. **(8 points)** For the following questions a - d, determine the outputs if the given statements are executed. Assume the below array has been declared and initialized.

```
int[][] ragged = {{1,2}, {3,4,5}, {6,7,8,9}};
```

a) **(2 points)** System.out.println(ragged[0][0]); _____
b) **(2 points)** System.out.println(ragged[2].length); _____
c) **(2 points)** System.out.println(ragged.length); _____
d) **(2 points)** System.out.println(ragged[1][2]); _____

S24 - 3.2b: Regular: ~57%, Conflict: ~58%

a) **(5 points)** Write a code snippet that declares a one-dimensional array of type *double* and size 5, and initializes it with any random numbers using Math.random(). You DO NOT need to include the headers (public class/public static void main).

b) **(10 points)** Write a code snippet that finds the minimum and maximum value in the array created in part a, and prints out the sum of these two values. You DO NOT need to include the headers (public class/public static void main).

S24 - 4.1: Regular: ~52%, Conflict: ~48%

1. **(10 points)** Write the Average program that reads an undetermined number of arguments from the command line as double values and displays the average of the inputs. Assume the inputs are guaranteed to be of type *double*. See the examples below:

- java Average 5.2 2.3 35.2 6.0 → 12.175
- java Average 5.0 6.0 → 5.5

```
public class Average
{
    Run | Debug
    public static void main(String[] args){
```

Java Basics/Arrays Answers

Answer F24 - 5.2:
int floor = (int) number;
return floor;

To round up:
int ceiling = (int)number + 1;
return ceiling;

Answers S24 - 3.2:
a) 1
b) 4

c) 3
d) 5

Answers S24 - 3.2b:
a)

```
double[] array = new double[5];
for (int i = 0; i < array.length; i++) {
        array[i] = Math.random();
}
```

b)

```
int min = array[0];
int max = array[0];
for (int i = 0; i < array.length; i++) {
        if (nums[i] > max) max = nums[i];
if (nums[i] < min) min = nums[i];
}
System.out.println(max + min);
```

Answer S24 - 4.1:

```
        int n = args.length;
        double sum = 0.0;

        // 5 points - reads an undetermined number of inputs from the command line
        for ( int i = 0; i < n; i++ ) {
            sum += Double.parseDouble(args[i]); // 2 points - use parseDouble to read an input
        }
        // System.out.println is also correct
        // 3 points - displays the average
        StdOut.print(sum/n);
    }
}
```

Final Exam

Note: You don't have to complete all of the questions! Focus on the topics you need the most help with and navigate using the document tabs on the left-hand side!

Largest Weaknesses for Final Exam (last updated S25):

1. OOP
2. Algorithms
3. Midterm 2 Content

Algorithm Visualizer: https://visualgo.net/en/sorting

Algorithms (including sorting, searching, and space/time complexity)

S25 - 4.2: Regular ~18%, Conflict ~24%

```java
public static int knuth(int[] vals) {
    int a = 0;
    for(int i = 0; i < vals.length; i++) {
        for(int j = 0; j < vals.length; j++) {
            if (i != j && vals[i] == vals[j]) {
                a++;
            }
        }
    }
    return a;
}
```

2. **(5 points)** Assuming that each int requires 4 bytes of storage, what is the space/memory requirement for the knuth method (provided in the above table)? Express your answer in terms of n.

Answer:

S25 - 4.3: Regular a) ~53% b) ~41% c) ~70%, Conflict a) ~70% b) ~40% c) ~81%

3. **(10 points)** Consider the following snippet of code:

```
15  for(int i = 0; i < n; i ++) {
16      System.out.println();
17      for(int j = 0; j <= i; j++) {
18          System.out.print("*");
19      }
20  }
```

a. **(3 points)** If n = 4, what will be printed?

b. **(5 points)** In terms of n, how many times will line 18 be executed?

c. **(2 points)** What is the runtime efficiency (using Big O notation) of this program?

S25 - 5.5: Regular ~27%, Conflict ~34%

5. **(4 points)** How many element-to-element comparisons are made to merge these two sorted subarrays together using the merge process from mergesort?

52	63	88	91

58	67	75	99

F24 - 5.2: Regular ~49%, Conflict ~60%

2. **(9 points)** Using the binary search algorithm seen in lecture, which array indices are investigated when running the algorithm to search for 17 in the following array? Write the indices in the order in which they are investigated.

index	0	1	2	3	4	5	6	7
a	2	3	7	9	11	17	20	41

F24 - 4.1: Regular ~58%, Conflict ~63%

Regular:

Problem 4 – Algorithm Complexity Analysis (30 points)

1. **(12 points - 2 points each)** Give the most efficient Big O (the fastest) for the worst-case scenario of the following algorithms.

Algorithm	Big-O Complexity
Setting the value of the last element in a 1D array of integers to 0	
Inserting an element to the beginning of an array	
Searching for a target value in a sorted 1D array of integers	
Counting the number of 0s (zeros) in an *n x n* 2D array of integers	
Summing the values on the diagonal of an *n x n* 2D array	
Finding the minimum value in a 1D array that is sorted in ascending order	

Conflict:

Problem 4 – Algorithm Complexity Analysis (30 points)

1. **(12 points - 2 points each)** Give the most efficient Big O (the fastest) for the worst-case scenario of the following algorithms.

Algorithm	Big O Complexity
Finding the minimum value in a 1D array that is sorted in descending order	
Setting the value of the middle element in a 1D array of integers to 0	
Inserting an element to the beginning of an array	
Searching for a target value in an ordered 1D array of integers	
Finding the largest value in an *n x n* 2D array of integers	
Summing the values in the last row of an *n x n* 2D array	

F24 - 5.6/5.7: Regular ~58/59%, Conflict ~53/63%

6. **(4 points)** Which sorting algorithm is the most efficient (fastest) to use (insertion, selection, or mergesort), if you know that the array is almost sorted, meaning that there are only a few items out of their final position. Succinctly explain why the algorithm you chose is better than the other two for this scenario.

7. **(3 points)** Which sorting algorithm is the most efficient (fastest) to use (insertion, selection, or mergesort), if you don't know anything about the array that needs to be sorted. Succinctly explain why the algorithm you chose is better than the other two for this scenario.

Algorithms Answers

Answer S25 - 4.2:

2. **(5 points)** Assuming that each int requires 4 bytes of storage, what is the space/memory requirement for the knuth method (provided in the above table)?
[+5 points] 4n + 12
[+2.5 points] 4n
[+2.5 points] 12

Answer S25 - 4.3

a. **(3 points)** If n = 4, what will be printed?
[+3 points]
*
**
***
****

b. **(5 points)** In terms of n, how many times will line 18 be executed?
[+5 points] 1 + 2 + … + (n-2) + (n-1) + n
[+5 points] $(n^2 + n)/2$
[+2 points] $n^2 + n$ (partial credit)
[+2 points] $n^2 / 2$ (partial credit)

c. **(2 points)** What is the runtime efficiency (using Big O notation) of this program?
[+2 points] $O(n^2)$

Answer S25 - 5.5: 7

Answer F24 - 5.2: 4, 6, 5

Answer F24 - 4.1 Regular/Conflict Respectively:

Algorithm	Big O complexity
Setting the value of the last element in a 1D array of integers to 0	O(1)
Inserting an element to the beginning of an array	O(n)
Searching for a target value in a sorted 1D array of integers	O(log n)
Counting the number of 0s (zeros) in an *n x n* 2D array of integers	$O(n^2)$
Summing the values on the diagonal of an *n x n* 2D array	O(n)
Finding the minimum value in a 1D array that is sorted in ascending order	O(1)

Algorithm	Big O complexity
Finding the minimum value in a 1D array that is sorted in descending order	O(1)
Setting the value of the middle element in a 1D array of integers to 0	O(1)
Inserting an element to the beginning of an array	O(n)
Searching for a target value in an ordered 1D array of integers	O(log n)
Finding the largest value in an *n x n* 2D array of integers	$O(n^2)$
Summing the values in the last row of an *n x n* 2D array	O(n)

Answer F24 - 5.6
2 points - Insertion sort.
2 points - Because insertion sort will run close to O(n) while selection sort will run in O(n^2) and
mergesort will run in O(n log n).

Answer F24 - 5.7
1 point - Mergesort
2 points - Because insertion sort and selection sort can run in O(n^2)

Midterm 2 Content

[S25 - 1.2a / 1.2b: Regular ~72%/~30%, Conflict ~35%/~34%]

2. **(10 points)** Consider the following piece of code that is stored in a file named IO_Final.java and the following text file named FinalString.txt.

```
public static void main(String[] args){
  int numEntries = Integer.parseInt(args[0]);
  int sum = 0;
  for (int i = 0; i < numEntries;i++){
      int n = StdIn.readInt();
      sum += n;
  }
  StdOut.println(sum);

  int stringLength = 0;
  while(! StdIn.isEmpty()){
      String s = StdIn.readString();
      int len = s.length();
      stringLength += len;

  }
  StdOut.println(stringLength);
}
```

FinalString.txt

```
4
5
6
8
dog
bear
cat
```

a) What would be printed if the Java program above was executed with the following command and input?

Output:

java IO_Final 3
1
2
3
Hi
Bye
No

b) What would be printed if the Java program above was executed with the following command?
java IO_Final 3 < FinalString.txt

Output:

S24 - 1.2: Regular ~22%

2. **(8 points)** What is the output of the command 'java TestingFunctions 2'?

```
public class TestingFunctions {

    Run | Debug
    public static void main(String[] args) {
        int a = Integer.parseInt(args[0]);
        functionA(a);
        System.out.println(a);
    }

    public static void functionA (int a) {
        a = a * a;
    }
}
```

Output:

S24 - 1.3b: Regular ~34%

3. **(15 points)** Consider the following Java functions.

```
public static int halve1 ( int x ) {
    x = x / 2;
    return x;
}
public static void halve2 ( int[] a ) {
    int n = a.length;
    for ( int i = 0; i < n; i++ ) {
        halve1(a[i]);
        a[i] = halve1(a[i]);
    }
}
```

```
public static void halve3 ( int[] a
    int n = a.length;
    int[] b = new int[n/2];
    for ( int i = 0; i < n/2; i++ )
        b[i] = a[i];
    }
    a = b;
}
```

a) **(7.5 points)** What are the contents of the array a[] after the execution of the following three lines of code? *Write your answer in the box below.*

```
int[] a = { 16, 32, 48, 64 };
halve2(a);
halve2(a);
```

b) **(7.5 points)** What are the contents of the array a[] after the execution of the following three lines of code? *Write your answer in the box below.*

```
int[] a = { 16, 32, 48, 64 };
halve3(a);
halve3(a);
```

Midterm 2 Content Answers

S25 - 1.2

a. What would be printed if the Java program above was executed with the following command and input:

java IO_Final 3
1
2
3
Hi
Bye
No
<control D> or <control C> to end input

Answer:
6
7

5 points (2.5 each)

b. What would be printed if the Java program above was executed with the following command and input:

java IO_Final 3 < FinalString.txt

Answer:
15
11

5 points (2.5 each)

Answer S24 - 1.2: 2

Answer S24 - 1.3a/1.3b:

a) **(7.5 points)** What are the contents of the array a[] after the execution of the following three lines of code? *Write your answer in the box below.*

```
int[] a = { 16, 32, 48, 64 };
halve2(a);
halve2(a);
```

4, 8, 12, 16

b) **(7.5 points)** What are the contents of the array a[] after the execution of the following three lines of code? *Write your answer in the box below.*

```
int[] a = { 16, 32, 48, 64 };
halve3(a);
halve3(a);
```

16,32,48,64

Recursion

S25 - 2.4: Regular ~54%, Conflict ~64%

4. **(11 points)** Given an array of integers, write a **recursive** function that computes if the array contains a 6. The function will return **true** if a 6 is present, and **false** otherwise. The function header is given for you. The first call for each test will pass in 0 as the index.

- array6([1, 6, 4], 0) returns true
- array6([2, 5], 0) returns false

```
public static boolean array6 (int[] nums, int index)
```

Recursion Answers

S25 - 2.4

```
if (index == nums.length)    //3 points correct base case
   return false; //1 point return false for base case
 if (nums[index] == 6) //3 points checking for 6
   return true; //1 point return true if 6 is found
```

OOP

S25 - 3.1f: Regular ~58%, Conflict ~75%

1. **(22 points)** A pet adoption facility is very attentive to the pets in its care. The facility keeps track of the pet's energy and perceived happiness! In this question, you will be creating a data type called Pet. The class structure is provided below for you. Every Pet object has a name, a species, and ratings for happiness and energy. You will complete the sections marked by the line "**WRITE YOUR CODE HERE**". The code you are to write is described in parts a – f below.

```
public class Pet{

    private String name;
    private String species;
    private int happiness;
    private int energy;
```

f) **(4 points)** Write a method **toString** that returns the representation of the **Pet** object as a String. The representation should include only the pet's name and species. Include the method header. The structure of the String is up to you.

```
// f) toString - Include the method header.
//WRITE YOUR CODE HERE
```

S25 - 3.2: Regular all parts a) ~45%, b) ~41%, c) ~48%, d) ~49%, e) 27%; Conflict all parts a) ~53%, b) ~53%, c) ~46%, d) ~58%, e) 79%

2. (13 points) The adoption facility runs a client program, PetTester, to check on the pets. Complete the code where indicated.

a) **(2 points)** Create an array called *pets* that contains *numPets* Pet objects.
b) **(3 points)** Create a new Pet object by calling the constructor with the information read from the text file and assign it to the ith element of the array.
c) **(2 points)** Write a line of code that will simulate playing with the pet at index 0.
d) **(2 points)** Write a line of code that will simulate feeding the pet at index 1.
e) **(4 points)** In the for-each loop given, print out each pet's information and each pet's status.

```
public class PetTester{

    Run | Debug
    public static void main(String[] args){
        int numPets = Integer.parseInt(args[0]);
        //a - array creation - WRITE YOUR CODE HERE



    for(int i = 0; i < pets.length; i++){
        String name = StdIn.readString();
        String species = StdIn.readString();
        int happiness = StdIn.readInt();
        int energy = StdIn.readInt();
        //part b - pet object creation - WRITE YOUR CODE HERE



    } //end of for-loop

     //part c - method call to play with pets[0] - WRITE YOUR CODE HERE



     //part d - method call to feed pets[1] - WRITE YOUR CODE HERE



for (Pet p : pets){
    //part e - print pet info. and status - WRITE YOUR CODE HERE




}//end for loop
```

OOP Concepts

S25 - 3.1f

```
// f) toString - Include the method header.
//WRITE YOUR CODE HERE
```

```
public String toString(){
    String s = this.name + ": " + this.species;
    return s;
}
```

1 pt header
1 pt for each attribute
1 pt for return

S25 - 3.2 all parts

```
public class PetTester{
   public static void main(String[] args){
     int numPets = Integer.parseInt(args[0]);
     CREATE AN ARRAY OF numPets Pet objects: WRITE CODE HERE.
       Pet[ ] pets = new Pet[numPets];                    (2 points)

       //This loop reads info about pet
     for(int i = 0; i < pets.length; i++){
       String name = StdIn.readString();
       String species = StdIn.readString();
       int happiness = StdIn.readInt();
       int energy = StdIn.readInt();

            WRITE CODE HERE TO create a new Pet object and assign it to the ith element of
            the array.

            pets[i] = new Pet(name,species,happiness,energy);      (3 points) //order should
            match constructor

     }

       WRITE A method CALL TO PLAY WITH pets[0];

                pets[0].playWithPet();                                    (2 points)

           WRITE A method CALL TO FEED pets[1].

            pets[1].feedPet();                                        (2 points)
     Complete the Java Loop below to complete the task given:
   for (Pet p : pets){
         WRITE CODE HERE to print pet info (name and species).
         System.out.println(p); OR System.out.println(p.toString);
                (2 points)

         WRITE CODE HERE to print pet status
         p.printPetStatus()                                           (2 points)

     } //end for loop
}//end class
```