

CS 1324 Spring 2021 Homework 7 Methods

Jordan McFadden

TOTAL POINTS

20 / 20

QUESTION 1

Question 1 4 pts

1.1 Question 1a 2 / 2

- ✓ - **0 pts** Correct
- **0.5 pts** Minor mistake
- **2 pts** Not answered
- **1 pts** Partial
- **2 pts** Incorrect
- **2 pts** Late

1.2 Question 1b 2 / 2

- ✓ - **0 pts** Correct
- **0.5 pts** Minor mistake
- **2 pts** Not answered
- **1 pts** Partial
- **2 pts** Incorrect
- **2 pts** Late

QUESTION 2

2 Question 2 5 / 5

- ✓ - **0 pts** Correct
- **0.5 pts** Minor mistake
- **1 pts** No arguments listed
- **1 pts** Partial
- **1 pts** Parameters incorrect
- **2 pts** partial
- **4 pts** Partial
- **5 pts** Incorrect/ Not answered/ Late submission

QUESTION 3

Question 3 9 pts

3.1 Question 3a 3 / 3

- ✓ - **0 pts** Correct
- **0.5 pts** Partial

- **1 pts** Partial

- **3 pts** No answer/ Late

3.2 Question 3b 3 / 3

- ✓ - **0 pts** Correct
- **0.5 pts** Partial
- **1 pts** Partial
- **3 pts** No answer/ Late

3.3 Question 3c 3 / 3

- ✓ - **0 pts** Correct
- **0.5 pts** partial
- **1 pts** partial
- **3 pts** No answer/ Late

QUESTION 4

4 Question 4 2 / 2

- ✓ - **0 pts** Correct
- **0.5 pts** Partial
- **1 pts** Incorrect
- **2 pts** Not answered/ Late
- **1.5 pts** No method defined

Homework 7: Methods

CS 1323/4 Spring 2021

Name: Jordan McFadden

Student ID (usually 112-XXX-XXXX or 113-XXX-XXXX): 113502650

1. (4 points; 2 points each)

Suppose we have the method defined below:

```
public static int countInRangeInclusive(int low, int high, int sentinel, String
prompt, Scanner keyboard)
{
    int count = 0;

    // Priming read
    System.out.println(prompt);
    int entry = keyboard.nextInt();
    keyboard.nextLine();

    // Sentinel controlled loop
    while (entry != sentinel)
    {
        // Increment count if the entry is between low and high inclusive
        if (entry >= low && entry <= high)
            count = count + 1;

        //Priming read
        System.out.println(prompt);
        entry = keyboard.nextInt();
        keyboard.nextLine();
    }

    return count;
}
```

a) Write a complete main program that calls the method above to print to the console the number of values entered at the keyboard that are between 5 and 10 (inclusive) until a -1 is entered. Because the Scanner is a parameter, it will need to be declared and constructed in the main program. Write your own prompt.

```
import java.util.Scanner
public class method
{
    public static void main(String[] args)
    {
        Scanner keyboard = new Scanner (System.in);
        System.out.print(countInRangeInclusive (5,10,-1, "Pick a number between (1-10)
and (-1) to exit", keyboard));
    }
}
```

b) Write a complete main program that calls the method above to print to the console the number of values entered at the keyboard that are between 1 and 100 (inclusive) until a 0 is entered. Because the Scanner is a parameter, it will need to be declared and constructed in the main program. Write your own prompt.

```
import java.util.Scanner
public class method
{
    public static void main(String[] args)
    {
        Scanner keyboard = new Scanner (System.in);
        System.out.print(countInRangeInclusive (1,100,0,"Pick a number between (1-1000)
        and (0) to exit", keyboard));
    }
}
```

2. (5 points; 1 point each line) In the countInRangeInclusive method from problem 1, show all identifiers that fall into each category. Some lines may be blank.

Parameter(s)	int low, int high, int sentinel, String prompt, Scanner keyboard
Arguments(s)	(5,10,-1,"Prompt",keyboard) (1,100,0,"Prompt",keyboard)
Local variables	count, entry
Return type	int
Method name	countInRangeInclusive()

3. (9 points) Write ***the signatures*** for each of the methods described below. **You do not need to write the method.** In fact, we do not know how to write some of these methods yet (but we will, I promise!).

a) The method will take three integer values and produce a String that has these values separated by spaces.

Example: If the method is given 5, 8, and 12, it will produce "5 8 12".

```
String spacedIntegers(int firstNum, int secondNum, int thirdNum);
```

b) The method will take three integer values, and return the one that is in the middle. The fancy math word for middle is median.

Example: If the method is given 5, 8 and 12, 8 will be returned. If the method were given 5, 5, and 5, 5 would be returned.

```
int median(int a, int b, int c)
```

c) The method will take two integer values and return whether or not one value is a factor of the other.

Example: If the method were given 8 and 12, it would return false. If the method were given 24 and 8, it would return true.

```
boolean isFactor(int x, int y)
```

4. (2 points) The program below takes a given number of integers entered from the console, allows the user to enter them, and finds the largest entered value.

Rewrite this program to use a method. Your main program should declare, construct, and close the Scanner. It should also call a method that does the work that is done between lines 15 and 29 in the code below. In the end, your main program should be about five lines long. The prompt inside the while loop should be a parameter (there are other parameters).

The box for your answer is on the next page.

```
import java.util.Scanner;

public class Largest
{
    // Find the largest value of a given number of values entered at the keyboard
    public static void main(String[] args)
    {
        Scanner keyboard = new Scanner(System.in);

        System.out.println("Enter the number of values to be entered");
        int numberValues = keyboard.nextInt();
        keyboard.nextLine();

        // Enter the values and find the largest // Line 15
        int count = 0;
        int max = Integer.MIN_VALUE; // by starting with the smallest value,
        // we know everything else is larger--this is a constant in the API
        while (count < numberValues)
        {
            System.out.println("Enter the next value");
            int value = keyboard.nextInt();
            if (value > max)
            {
                max = value;
            }

            count = count + 1;
        } // Line 29

        System.out.println("The maximum is " + max);
    }
}
```

```
import java.util.Scanner;
public class Largest
{
    public static void main(String[] args)
    {
        Scanner keyboard = new Scanner(System.in);
        System.out.println("Enter the number of values to be entered");
        int numberValues = keyboard.nextInt();
        keyboard.nextLine();
        System.out.println("The maximum is " + (max(keyboard,"Enter the next
value.", numberValues)));
    }
    public static int max(Scanner keyboard, String prompt, int numOfValues)
    {
        // Enter the values and find the largest // Line 15
        int count = 0;
        int max = Integer.MIN_VALUE; // by starting with the smallest value,
        // we know everything else is larger--this is a constant in the API
        while (count < numOfValues)
        {
            System.out.println(prompt);
            int value = keyboard.nextInt();
            if (value > max)
            {
                max = value;
            }
            count = count + 1;
        }
        return max;
    }
}
```