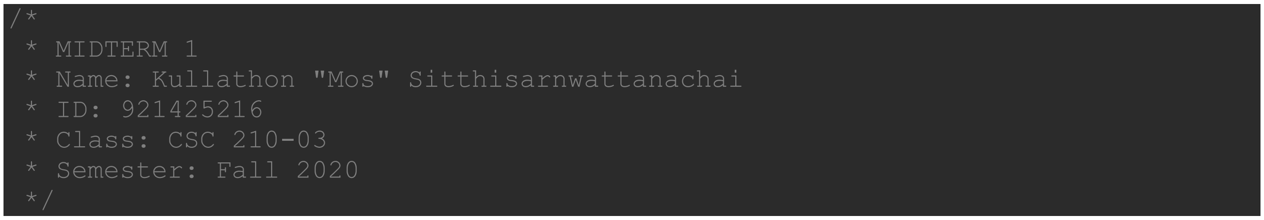


```
/*
 * MIDTERM 1
 * Name: Kullathon "Mos" Sitthisarnwattanachai
 * ID: 921425216
 * Class: CSC 210-03
 * Semester: Fall 2020
 */
```

Midterm 1

1. [7 points] Complete the code below to achieve the expected result.

Expected Result:

6.00 3.00 1.50 0.75 0.38 0.19

Code:

```
double num = 6.0;
for (int i = 1; i<=6; i++) {
       System.out.printf("%.2f ", num);
        num /= 2;
}
```

2. [7 points] What will be the output of the following code?

```
for ( i=0; i<10; i++) {
  if (i > 6)
     continue;
}
System.out.println(i);
```

A. 6

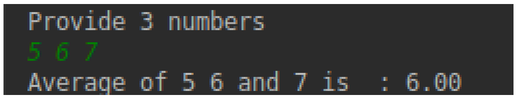
B. 7

C. 10

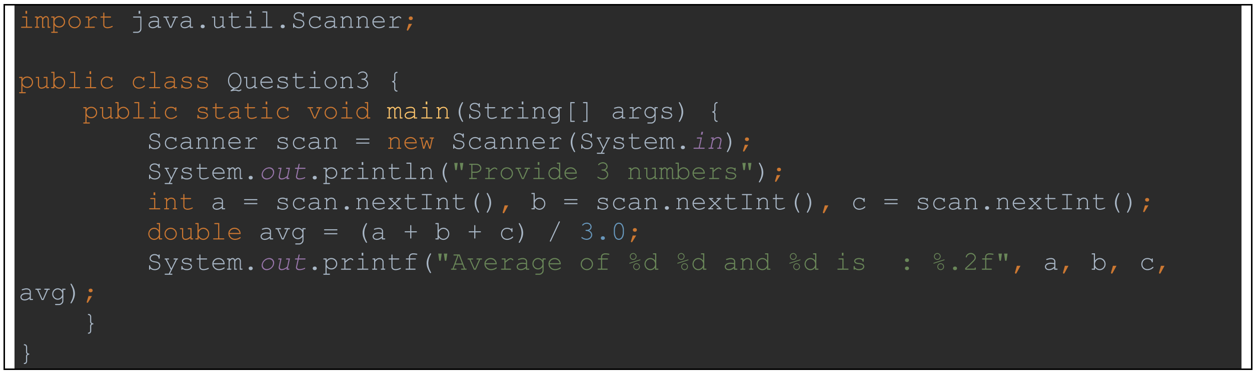
D. 11

E. Compilation Fails – variable "i" has not been declared or initialized.

3. [7 points] Write code that prompts the user for 3 integer values, calculate the average and outputs exactly the following:



```
Provide 3 numbers
5 6 7
Average of 5 6 and 7 is  : 6.00
```



```
import java.util.Scanner;

public class Question3 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Provide 3 numbers");
        int a = scan.nextInt(), b = scan.nextInt(), c = scan.nextInt();
        double avg = (a + b + c) / 3.0;
        System.out.printf("Average of %d %d and %d is  : %.2f", a, b, c,
avg);
    }
}
```

4. [7 points]

a. Name 3 types of modifiers used in a method header, provide examples of each, and explain how each are used

a1. **Access modifiers** – dictates the access level of the method, i.e., specifies if the method can be accessed by members outside of the class. Two common ones are “public” and “private.” The public modifier allows the method to be accessible for all classes and the private modifier only allows the method to be accessible within the class itself. For example, the main method is declared as public, as you can see from its header: “**public** static void main(String[] args)”. Methods can be declared as private by replacing the “public” keyword as “private.” This may be useful for methods that are only concerned with the class itself and can prevent it from being accessed unnecessarily outside the context of the class itself.

a2. **Instance modifiers** – dictates whether the method is static or non-static. Static methods are methods that can be called without creating an instance of the class itself. They are methods that can be called by referencing the class itself. Non-static methods are the opposite. They are methods that must be called from an instance. Within the class, non-static methods cannot be called from static methods (for example the main method). For example, the main method is a static method and is declared with the header: “public **static** void main(String[] args)” the static (boldened) indicates that the method is of a static type and can be called without the creation of its instance. A non-static method would omit the static keyword altogether.

a3. **Return type** – dictates if and what the method will return. “void” is used to indicate that the method will not return a value upon execution and so a return statement is optional in the method. Whereas, if a method were to return something, it must declare its *return type* in its header (otherwise “void”). If a return type is specified (i.e., not “void”), then the method must have a return statement (that returns a data of the specified type). For example, if a method were to return a integer, then its header would look something like: “public static **int** doSomething()” the int (boldened) specifies that the method, doSomething, would return a data that is an integer.

b. Name 5 primitive data types

boolean, char, int, long, float, double

5 [8 points] a. What is the output of x at A, B, and C

x = **1** at A

x = **1** at B

n = **2** at C // NOTE: Correction made: instead of “x = … at C” it is “n = … at C” as clarified on Discord

```
public class Increment {

        public static void main(String[] args) {
        int x = 1;
        System.out.println("Before the call, x is " + x);// A
        increment(x);
        System.out.println("After the call, x is " + x); // B
        }

        public static void increment(int n) {
        n++;
        System.out.println("n inside the method is " + n); // C
        }
}
```

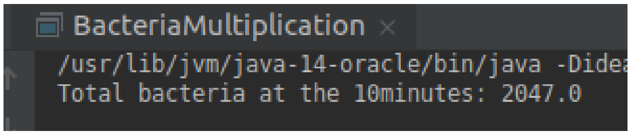
5b. How would you modify the above code so that x is increment by 1 at B

On the same line, pre-increment the x, i.e., “System.out.println(“After the call, x is “ + **++**x); // B”. Method left unchanged on purpose as to not modify its behavior.

6. [7 points] We have a bacteria, the bacteria splits into 2, every minute. You can express the number of bacteria created in a series of number 1, 2, 4, 8, 16, etc. At the start, the bacteria count is 1, the 2nd minute the 1 bacteria splits into 2 so the sum is 3 (i.e. 2+1), the 3rd minute, the 2 will split again and total the count is 4 + 2 + 1, etc.

Please use a loop and a Math function to get the sum of all the bacteria after 1 hour.

Example output



```
/usr/lib/jvm/java-14-oracle/bin/java -Dide
Total bacteria at the 10minutes: 2047.0
```

```
public class Question6 {
    public static void main(String[] args) {
        double bacTotal = 1;
        int totalMinutes = 60;

        for (int i = 1; i <= totalMinutes; i++) {
            bacTotal += Math.pow(2, i);
        }

        System.out.printf("Total bacteria at the %d minutes: %.1f",
totalMinutes, bacTotal);

    }
}
```

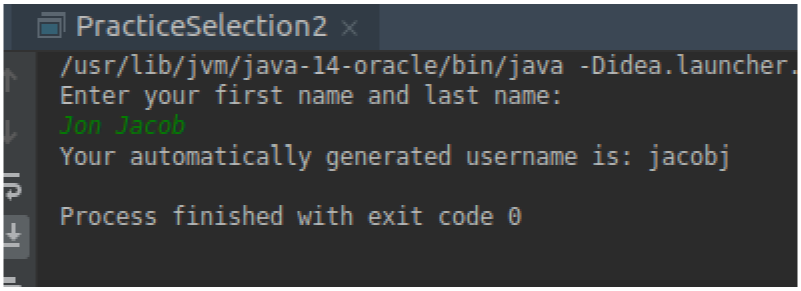
7. [7 points] Complete the following code to generate a username that is constructed out of last name and first letter of the last name.

We have prompted the user to enter their First Name and Last Name, for example Jon Jacob.

- Save an automatically generated username in a String variable that is compiled by concatenating the last name first then the first character of the first name, and all will be in lower case. So an entry from the user of “Jon Jacob”, will have a username automatically generated as “jacobj”

- Print out the user name to the console as “Your automatically generated username is jacobj”

```
Scanner scan = new Scanner(System.in);

System.out.println("Enter your first name and last name, separated by a space: ");

String first = scan.next();
String last = scan.next();
System.out.printf("Your automatically generated username is: %s%s", last.toLowerCase(),
first.toLowerCase().charAt(0));
```



```
/usr/lib/jvm/java-14-oracle/bin/java -Didea.launcher.
Enter your first name and last name:
Jon Jacob
Your automatically generated username is: jacobj

Process finished with exit code 0
```

8. [7 points]

a. Each method in a class must have a different name. True/False?

False.

b. What is method overloading?

Creating multiple methods with the same name but with different signatures. This can be done either by having different number of arguments, or have arguments of different types.

9. [7 points] What are the values of DAYS_IN_WEEK for each row number?

```
1  final int DAYS_IN_WEEK = 7;
2 DAYS_IN_WEEK += 7;
3 System.out.print(DAYS_IN_WEEK + 7);
```

Value is 7 for all rows because it is final. Java won’t even let you increment in the second line because the var is final.

10. Please trace the following values for the below program and identify the data type

a. length at row #2 : value **10** datatype is **int**

b. temp at row # 7 when (isDigit(temp)) evaluates to true : value **‘5’** datatype is **char**

c. username.substring(i) at row # 8: value **“5”** datatype is **String**

d. i at row #8: value **8** datatype is **int**

e. userDigit at row#12: value **55** datatype is **int**

```
1 String username = "monalisa55";
2 int length = username.length();
3 char temp;
4 int userDigit=0;
5 for (int i = 0; i< length; i++){
6    temp = username.charAt(i);
7    if (isDigit(temp)){
8        userDigit = Integer.parseInt((username.substring(i)));
9        break;
10   }
11 }
12 System.out.println(userDigit);
```

11.[7 points] How would you modify this code into a do-while loop?

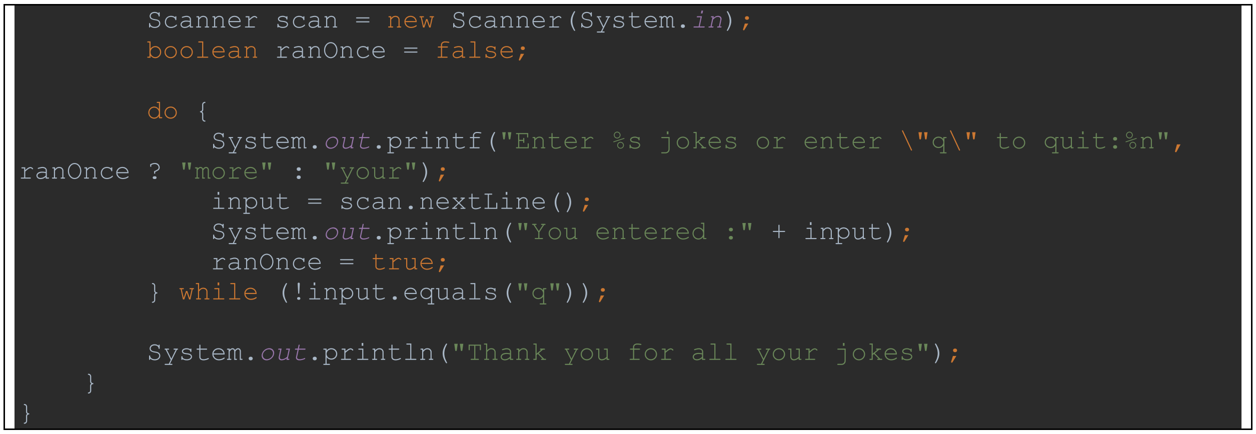
```
System.out.println("Enter your jokes or enter "q" to quit:")
String input = scan.nextLine(); // say the user enters 6
System.out.println("You entered :" + input);

while (input != "q" ) {
  System.out.println("Enter more jokes or enter "q" to quit:")
  input = scan.nextLine();
  System.out.println("You entered :" + input);

}
System.out.println("Thank you for all your jokes");
```

Note: there are several errors in the original code above:

- Scanner object “scan” not defined.
- Missing semicolon on several print statements.
- Unescaped quotes in strings.
- Not using equals() method for String comparison on while condition.

Below is the code with all the abovementioned errors fixed and the code written using do-while loop.

```
import java.util.Scanner;

public class Question11 {
    public static void main(String[] args) {
        String input;
```



```
        Scanner scan = new Scanner(System.in);
        boolean ranOnce = false;

        do {
            System.out.printf("Enter %s jokes or enter \"q\" to quit:%n",
ranOnce ? "more" : "your");
            input = scan.nextLine();
            System.out.println("You entered :" + input);
            ranOnce = true;
        } while (!input.equals("q"));

        System.out.println("Thank you for all your jokes");
    }
}
```

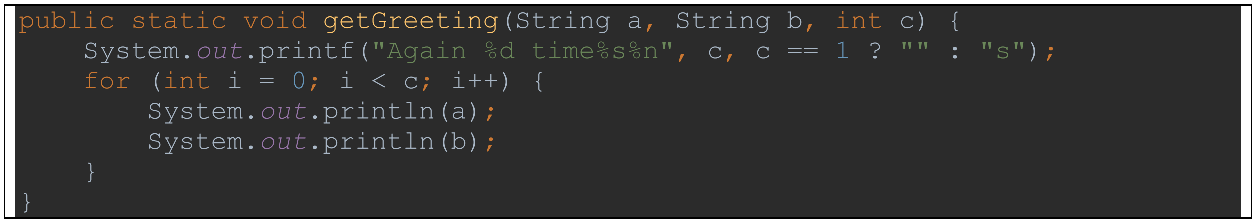
12. [7 points] Please write the method header that would output the following if main contains the following code:

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

  String hi = "Hi!";
  String bye = "Bye!";

  getGreeting(hi,bye,1);
  System.out.println();
  getGreeting(hi,bye,5);

}
```

Output

```
Again 1 time
Hi!
Bye!


Again 5 times
Hi!
Bye!
Hi!
Bye!
Hi!
Bye!
Hi!
Bye!
Hi!
```

```
Bye!
```



```java
public static void getGreeting(String a, String b, int c) {
    System.out.printf("Again %d time%s%n", c, c == 1 ? "" : "s");
    for (int i = 0; i < c; i++) {
        System.out.println(a);
        System.out.println(b);
    }
}
```

13. [7 points]

a. Please explain with examples what is the difference between high level programming language and low level programming language?

High-level languages are languages that are further from the nature of the machine – in other words, they are easier for humans to understand. The lowest level languages is the binary system, consisting of 0s and 1s. Higher level languages provides abstractions from the lower level ones so that it is easier to build programs.

b. Java is a low level language , True or False?

False. It is considered a high-level language.

14. [8 points] Identify the errors in this program and explain your rationale

a. logic error(s) on row #

row # 2 rationale : scanned value was never saved

row # 5/8 rationale : incorrect side for equality sign; should be overwritten if score is lower than the min bound or higher than max bound

row # 7/9 rationale : should be printing the bounds instead of the user-entered score, i.e., scoreMin and scoreMax, respectively

b. syntax error(s) on row #

row # 5/8 rationale : writtenScore not initialized; referenced before declaration

row # 6/9 rationale : writtenScore was never declared

row # 7 rationale : missing string concatenation "+"

row # 10 rationale : incorrect format parameter; should be "%d" not "$d"

```
1 System.out.println("Enter your written exam score" );

2 scan.nextInt();

// if smaller than 0 then assign to 0
// if larger than 500 assign to 500

3 int scoreMax = 500;
```

```
4 int scoreMin = 0;

5 if (writtenScore > scoreMin ){
6   writtenScore = 0 ;
7   System.out.println("Your written score is rewritten to : " , writtenScore );

8 }else if (writtenScore < scoreMax ){
9   writtenScore = 500 ;
10   System.out.printf("Your written score is rewritten to: $d" , writtenScore );

11 }// if score entered is between 0 and 500
12 else {
13  System.out.println("Your written score is " + writtenScore);
}
```

c. Please rewrite the code and fix the errors you've found

Minimal fixes only; inconsistencies and convention violations are left as-is.

```
import java.util.Scanner;

public class Question14 {
    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        System.out.println("Enter your written exam score");
        int writtenScore = scan.nextInt();

        // if smaller than 0 then assign to 0
        // if larger than 500 assign to 500

        int scoreMax = 500;
        int scoreMin = 0;

        if (writtenScore < scoreMin) {
            System.out.println("Your written score is rewritten to : " +
scoreMin);
        } else if (writtenScore > scoreMax) {
            System.out.printf("Your written score is rewritten to: %d",
scoreMax);
        } // if score entered is between 0 and 500
        else {
            System.out.println("Your written score is " + writtenScore);
        }

    }
}
```