Programming I - Assignment 3

Student Name: Evelyn Toledo Lally
Student Number: 22102020

1. Write a program that counts the number of digits in an integer entered by the user. The program should repeatedly ask for input and displays the number of digits the input integer has.

Code:

```
package Section3;

//importing scanner class to used to get user's input
import java.util.Scanner;

public class CountIntegers {

//Main method
public static void main String[] args) {

		// Declare and initialize variables
		int num = 0;

		// Show the program information to the user
		System.out.println("This program counts the number of digits in an integer entered by the user. Enter -1 to exit.");

		// Declare & create an instance of the Scanner class
		Scanner input = new Scanner(System.in);


		// Loop until user inputs -1
		while (num != -1) {
			// Request input from user
			System.out.println("Enter a number:");
			// Read in integer value and assign to variable
		    num = input.nextInt();
		    //change num to type string and assigns to variable length, then counts its length
			int length = String.valueOf(num).length();


			// Conditional Branching (If the user inputs any number apart from -1)
				if (num!= -1) {
					//Prints the length of the number inputed to the screen
				  System.out.println("Number of digits in " +num+ " is " +length);
				// and exit the program if the user inputs -1
				} else if (num == -1){
						// Terminates the program
						System.out.println("Program Terminating");

				}
```

```
            } // end of while loop

            // close input to avoid possible leak
            input.close();

        } // end main method

} // end class
```

2. Using the following UML diagram, you are required to:

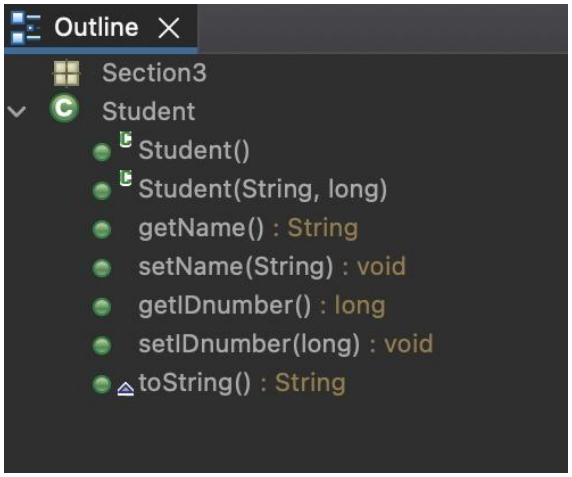
1. **Implement the Student class.**
    a. **In the instance variable box, the red square indicates that they are private**
    b. **In the methods box the c beside the Student indicates that the method is a constructor. In this example there are two Student constructors, one a default no argument constructor and the second a constructor that is expecting two arguments, a String and a long.**
    c. **toString():String – indicates a method called toString that returns an argument of type String. The method signature is public String toString()**

Code:

```
package Section3;

public class Student {

//Declare 2 private instance variables, but don't assign any value
        private String name;
        private long IDnumber;

//Constructor Student() with no arguments
        public Student() {
        }

//Constructor Student() passing 2 arguments types and names
        public Student(String name, long IDnumber) {
                this.name = name;
                this.IDnumber = IDnumber;
        }

//public getter & setter for each variables
        public String getName() {
                return name;
        }

        public void setName(String name) {
                this.name = name;
        }

        public long getIDnumber() {
                return IDnumber;
        }
```

```
    public void setIDnumber(long iDnumber) {
        IDnumber = iDnumber;
    }

//toString method will format how the student array will be returned
    public String toString() {
        return "Student [name=" + name + ", IDnumber=" + IDnumber + "]";
    }

} // end of class
```

Outline:



2. **Create a tester class which:**
    a. **creates an array to hold a collection of Student objects.**
    b. **adds a number of instances of Student to the array.**
    c. **uses a loop to traverse the array and display the contents.**

Code:

```
package Section3;

public class StudentTester {

//Main method
    public static void main(String[] args) {

        // Create new Student objects with 2 arguments (name and IDnumber)
        Student s1 = new Student("Bob", 1010);
        Student s2 = new Student("Ana", 1011);
        Student s3 = new Student("Jim", 1012);
        Student s4 = new Student("Fabs", 1013);
        Student s5 = new Student("Zeus", 1014);
        Student s6 = new Student("John", 1015);

        // Create an array to hold the collection of Student objects
        Student[] students = {s1, s2, s3, s4, s5, s6};

        // for loop will go through each array item, until it reaches the last item
        // and it will print each student parameters
        for(int i = 0; i < students.length; i++) {
            System.out.println(students[i]);

        } // end of for loop
```

```
    } // end of main method

} // end of class
```