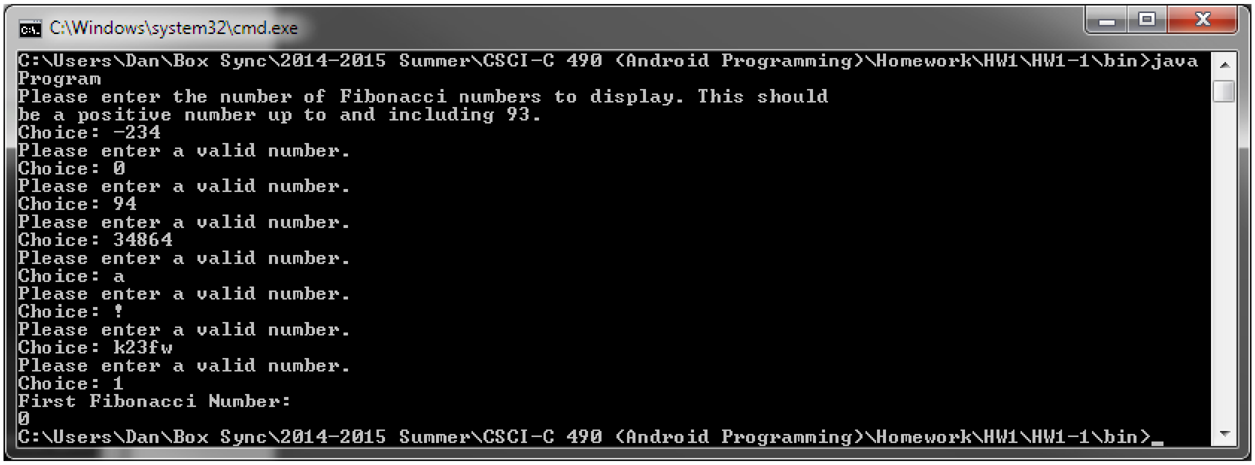
Name:	Dan Cassidy
Class:	CSCI-C 490, Mobile Application Development
Assignment:	Homework 1 Part 1
Date:	2015-07-09

```java
/*--------------------------------------------------------------------------------------------------
 * Author:      Dan Cassidy
 * Date:        2015-07-09
 * Assignment:  HW1-1
 * Source File: Program.java
 * Language:    Java
 * Course:      CSCI-C 490, Android Programming, MoWe 08:00
--------------------------------------------------------------------------------------------------*/


import java.util.Scanner;

/**
 * A small Java program to print out the first <i>n</i> (0 < <i>n</i> < 94) Fibonacci numbers.
 * @author Dan Cassidy
 */
public class Program
{
    /**
     * Entry point into the Java program.
     * @param args Command line arguments. <i>Not used.</i>
     * @return Nothing.
     */
    public static void main(String[] args)
    {
        // Define min and max.
        final byte NUM_MIN = 1;
        final byte NUM_MAX = 93;

        // Give the user a small description.
        System.out.println("Please enter the number of Fibonacci numbers to display. This should");
        System.out.println("be a positive number up to and including 93.");

        // Declare and prep variables for later use.
        byte numFib = 0;
        boolean validInput = false;
        Scanner consoleInput = new Scanner(System.in);

        // Loop while the user does not provide valid input.
        while (!validInput)
        {
            System.out.print("Choice: ");
            try
            {
                // Get console input.
                numFib = Byte.parseByte(consoleInput.nextLine());

                // Check for valid input and throw an exception if invalid.
                if (numFib < NUM_MIN || numFib > NUM_MAX)
                    throw new Exception();
                else
                    validInput = true;
            }
            catch (Exception ex)
            {
                System.out.println("Please enter a valid number.");
            }
        }

        // Print the Fibonacci numbers.
        System.out.println("First " + (numFib > 1 ? numFib + " " : "") + "Fibonacci Number" +
                (numFib != 1 ? "s": "") + ":");
        Fibonacci.calculateAndDisplay(numFib);
    }
}

```

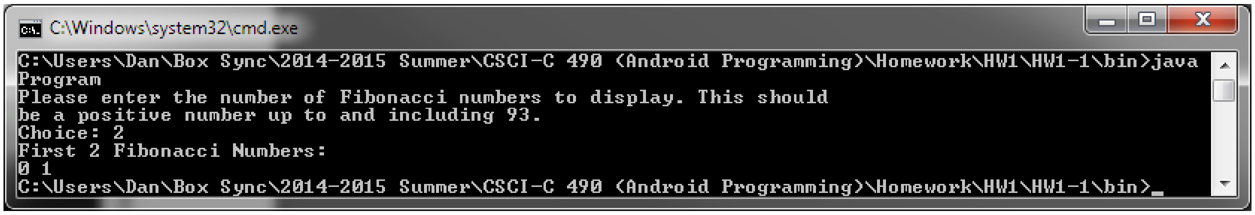
```
/*--------------------------------------------------------------------------------------------------
 * Author:      Dan Cassidy
 * Date:        2015-07-09
 * Assignment:  HW1-1
 * Source File: Fibonacci.java
 * Language:    Java
 * Course:      CSCI-C 490, Android Programming, MoWe 08:00
--------------------------------------------------------------------------------------------------*/

/**
 * Simple class dealing with Fibonacci numbers. Only handles up to the 93rd (F92) Fibonacci number
 * due to the fact the class only uses the long primitive type.
 * @author Dan Cassidy
 */
public class Fibonacci
{
    private final static byte F0 = 0;
    private final static byte F1 = 1;

    /**
     * Calculates and displays a given number of Fibonacci numbers.
     * @param sequenceLength The number of Fibonacci numbers to calculate and display.
     */
    public static void calculateAndDisplay(byte sequenceLength)
    {
        // Declare the requisite numbers.
        long fibA = F1, fibB = F0, fibSum = 0;

        // Display the first couple numbers in the sequence, if needed.
        if (sequenceLength > 0)
            System.out.print(F0 + " ");
        if (sequenceLength > 1)
            System.out.print(F1 + " ");

        // Iteratively calculate and display the rest of the Fibonacci sequence.
        for (int i = 2; i < sequenceLength; i++)
        {
            fibSum = fibA + fibB;
            fibB = fibA;
            fibA = fibSum;
            System.out.print(fibSum + " ");
        }
    }
}

```

Shows the program response to bad input, as well as the first Fibonacci number (edge case of 1).



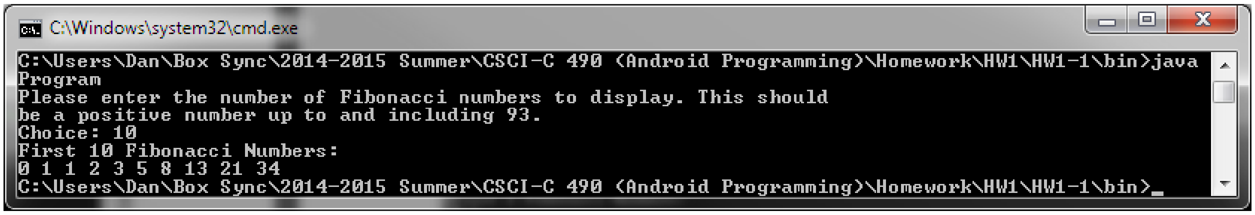
```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-1\bin>java
Program
Please enter the number of Fibonacci numbers to display. This should
be a positive number up to and including 93.
Choice: -234
Please enter a valid number.
Choice: 0
Please enter a valid number.
Choice: 94
Please enter a valid number.
Choice: 34864
Please enter a valid number.
Choice: a
Please enter a valid number.
Choice: !
Please enter a valid number.
Choice: k23fw
Please enter a valid number.
Choice: 1
First Fibonacci Number:
0
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-1\bin>_
```

Shows “normal choice of 2.



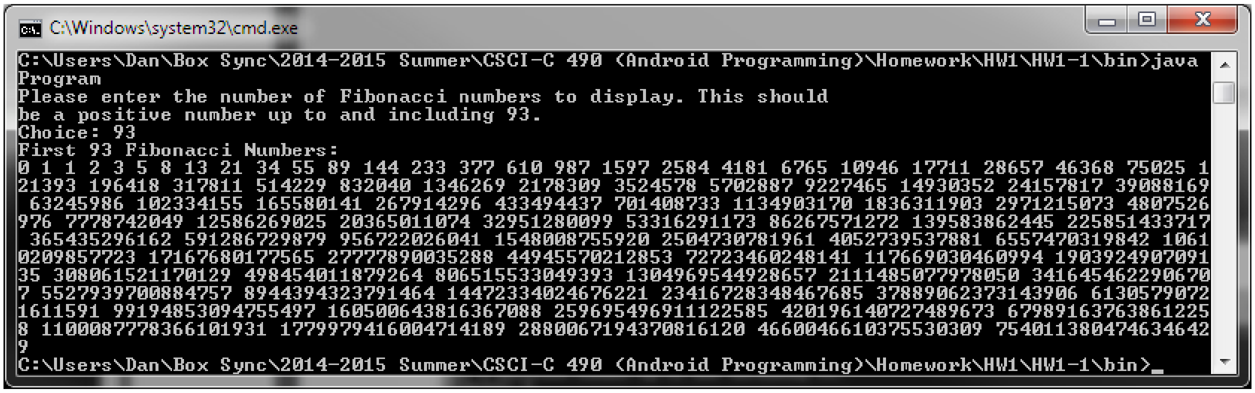
```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-1\bin>java
Program
Please enter the number of Fibonacci numbers to display. This should
be a positive number up to and including 93.
Choice: 2
First 2 Fibonacci Numbers:
0 1
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-1\bin>_
```

Shows “normal” choice of 10.



```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-1\bin>java
Program
Please enter the number of Fibonacci numbers to display. This should
be a positive number up to and including 93.
Choice: 10
First 10 Fibonacci Numbers:
0 1 1 2 3 5 8 13 21 34
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-1\bin>_
```

Shows the edge case of 93.



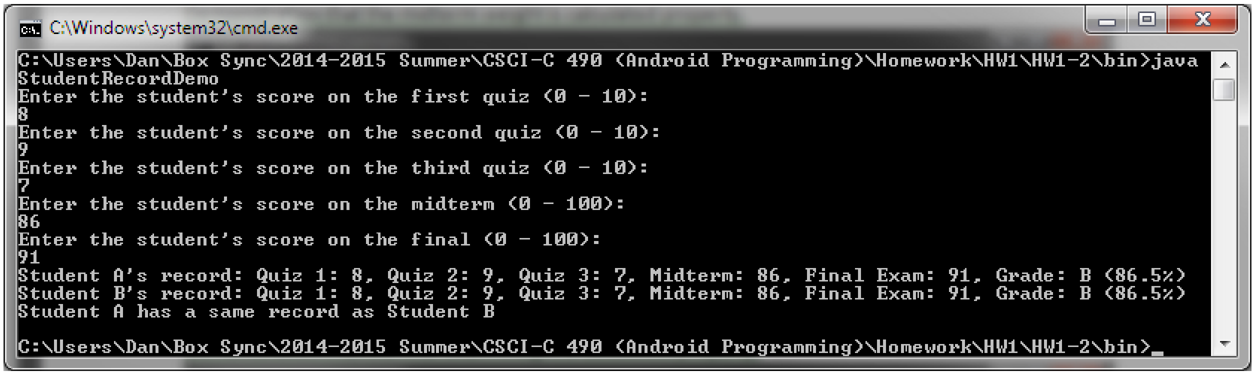
```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-1\bin>java
Program
Please enter the number of Fibonacci numbers to display. This should
be a positive number up to and including 93.
Choice: 93
First 93 Fibonacci Numbers:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 1
21393 196418 317811 514229 832040 1346269 2178309 3524578 5702887 9227465 14930352 24157817 39088169
 63245986 102334155 165580141 267914296 433494437 701408733 1134903170 1836311903 2971215073 4807526
976 7778742049 12586269025 20365011074 32951280099 53316291173 86267571272 139583862445 225851433717
 365435296162 591286729879 956722026041 1548008755920 2504730781961 4052739537881 6557470319842 1061
0209857723 17167680177565 27777890035288 44945570212853 72723460248141 117669030460994 1903924907091
35 308061521170129 498454011879264 806515533049393 1304969544928657 2111485077978050 341645462290670
7 5527939700884757 8944394323791464 14472334024676221 23416728348467685 37889062373143906 6130579072
1611591 99194853094755497 160500643816367088 259695496911122585 420196140727489673 67989163763861225
8 1100087778366101931 1779979416004714189 2880067194370816120 4660046610375530309 754011380474634642
9
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-1\bin>_
```

Name:	Dan Cassidy
Class:	CSCI-C 490, Mobile Application Development
Assignment:	Homework 1 Part 2
Date:	2015-07-09

```java
/*--------------------------------------------------------------------------------------------------
 * Author:      Dan Cassidy and Dr. Zhang
 * Date:        2015-07-09
 * Assignment:  HW1-2
 * Source File: StudentRecord.java
 * Language:    Java
 * Course:      CSCI-C 490, Android Programming, MoWe 08:00
--------------------------------------------------------------------------------------------------*/

/**
 * A small record of grades for a student.
 * @author Dan Cassidy
 * @author Dr. Zhang
 */
public class StudentRecord
{
    private int quiz1 = 0;
    private int quiz2 = 0;
    private int quiz3 = 0;
    private int midterm = 0;
    private int finalExam = 0;

    private double numericScore = 0.0D;
    private char letterGrade = '?';

    private final int A_GRADE = 90;
    private final int B_GRADE = 80;
    private final int C_GRADE = 70;
    private final int D_GRADE = 60;

    private final int NUMBER_OF_QUIZZES = 3;

    private final int QUIZ_MIN_SCORE = 0;
    private final int QUIZ_MAX_SCORE = 10;
    private final int MIDTERM_MIN_SCORE = 0;
    private final int MIDTERM_MAX_SCORE = 100;
    private final int FINAL_EXAM_MIN_SCORE = 0;
    private final int FINAL_EXAM_MAX_SCORE = 100;

    private final int QUIZ_WEIGHT = 25;
    private final int MIDTERM_WEIGHT = 35;
    private final int FINAL_EXAM_WEIGHT = 40;

    /**
     * Constructor.
     * @param quiz1 Student score for quiz 1.
     * @param quiz2 Student score for quiz 2.
     * @param quiz3 Student score for quiz 3.
     * @param midterm Student score for the midterm.
     * @param finalExam Student score for the final exam.
     */
    public StudentRecord(int quiz1, int quiz2, int quiz3, int midterm, int finalExam)
    {
        // Use this object's own mutators to set its instance variables, allowing verification
        // to happen in a single location.
        this.setQuiz1(quiz1);
        this.setQuiz2(quiz2);
        this.setQuiz3(quiz3);
        this.setMidterm(midterm);
        this.setFinalExam(finalExam);
    }

    // BEGIN GETTERS AND SETTERS -->
    public int getQuiz1()
    {
        return this.quiz1;
    }

    public void setQuiz1(int score)
    {
        if (score >= QUIZ_MIN_SCORE && score <= QUIZ_MAX_SCORE)
            this.quiz1 = score;
    }

    public int getQuiz2()
    {
        return this.quiz1;
    }

    public void setQuiz2(int score)
    {
        if (score >= QUIZ_MIN_SCORE && score <= QUIZ_MAX_SCORE)
            this.quiz2 = score;
    }

    public int getQuiz3()
    {
        return this.quiz1;
    }

    public void setQuiz3(int score)
    {
        if (score >= QUIZ_MIN_SCORE && score <= QUIZ_MAX_SCORE)
            this.quiz3 = score;
    }

    public int getMidterm()
    {
        return this.midterm;
    }

    public void setMidterm(int score)
    {
        if (score >= MIDTERM_MIN_SCORE && score <= MIDTERM_MAX_SCORE)
            this.midterm = score;
    }

    public int getFinalExam()
    {
        return this.finalExam;
    }

    public void setFinalExam(int score)
    {
        if (score >= FINAL_EXAM_MIN_SCORE && score <= FINAL_EXAM_MAX_SCORE)
            this.finalExam = score;
    }
    // <-- END GETTERS AND SETTERS

    /**
```

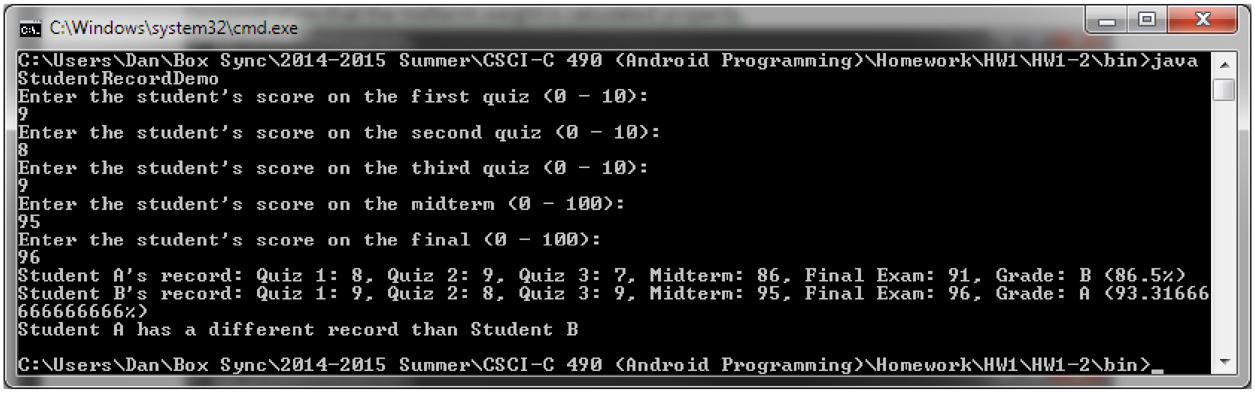
```java
     * This method calculates the numericScore based on the scores of the quizzes and exams.
     * @return Nothing.
     */
    public void computeNumericScore()
    {
        this.numericScore =
            (double)(quiz1 + quiz2 + quiz3) / (NUMBER_OF_QUIZZES * QUIZ_MAX_SCORE) * QUIZ_WEIGHT +
            (double)midterm / MIDTERM_MAX_SCORE * MIDTERM_WEIGHT +
            (double)finalExam / FINAL_EXAM_MAX_SCORE * FINAL_EXAM_WEIGHT;
    }

    /**
     * This method calculates the letterGrade based on the numberScore.
     * @return Nothing.
     */
    public void computeLetterGrade()
    {
        computeNumericScore();
        if (numericScore >= A_GRADE)
            letterGrade = 'A';
        else if (numericScore >= B_GRADE)
            letterGrade = 'B';
        else if (numericScore >= C_GRADE)
            letterGrade = 'C';
        else if (numericScore >= D_GRADE)
            letterGrade = 'D';
        else
            letterGrade = 'F';
    }

    /**
     * This method compares two StudentRecord objects. It will return true only if two objects
     * have same score for each of the quizzes and exams.
     * @param other Another StudentRecord object that will be compared against for equality.
     * @return boolean, showing whether the two student records are equal (true) or not (false).
     */
    public boolean equals(StudentRecord other)
    {
        if (other == null)
            return false;
        else
            return  (this.quiz1 == other.quiz1) &&
                    (this.quiz2 == other.quiz2) &&
                    (this.quiz3 == other.quiz3) &&
                    (this.midterm == other.midterm) &&
                    (this.finalExam == other.finalExam);
    }

    /**
     * This method returns a string representation of the data in the calling object.
     * @return A string representation of the StudentRecord object.
     */
    public String toString()
    {
        this.computeLetterGrade();
        return  "Quiz 1: " + this.quiz1 + ", " +
                "Quiz 2: " + this.quiz2 + ", " +
                "Quiz 3: " + this.quiz3 + ", " +
                "Midterm: " + this.midterm + ", " +
                "Final Exam: " + this.finalExam + ", " +
                "Grade: " + this.letterGrade + " (" +
                this.numericScore + "%)";
    }
}// StudentRecord

```

Demonstration that the equals() method finds two students equal if they have the same grades.



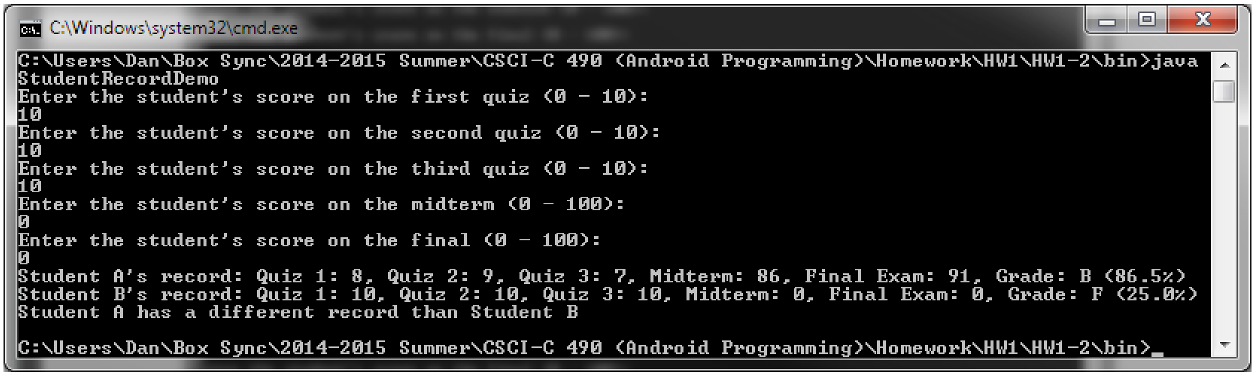
```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-2\bin>java
StudentRecordDemo
Enter the student's score on the first quiz (0 - 10):
8
Enter the student's score on the second quiz (0 - 10):
9
Enter the student's score on the third quiz (0 - 10):
7
Enter the student's score on the midterm (0 - 100):
86
Enter the student's score on the final (0 - 100):
91
Student A's record: Quiz 1: 8, Quiz 2: 9, Quiz 3: 7, Midterm: 86, Final Exam: 91, Grade: B (86.5%)
Student B's record: Quiz 1: 8, Quiz 2: 9, Quiz 3: 7, Midterm: 86, Final Exam: 91, Grade: B (86.5%)
Student A has a same record as Student B

C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-2\bin>_
```

Demonstration that the equals() method finds two students not equal if they have differing grades.



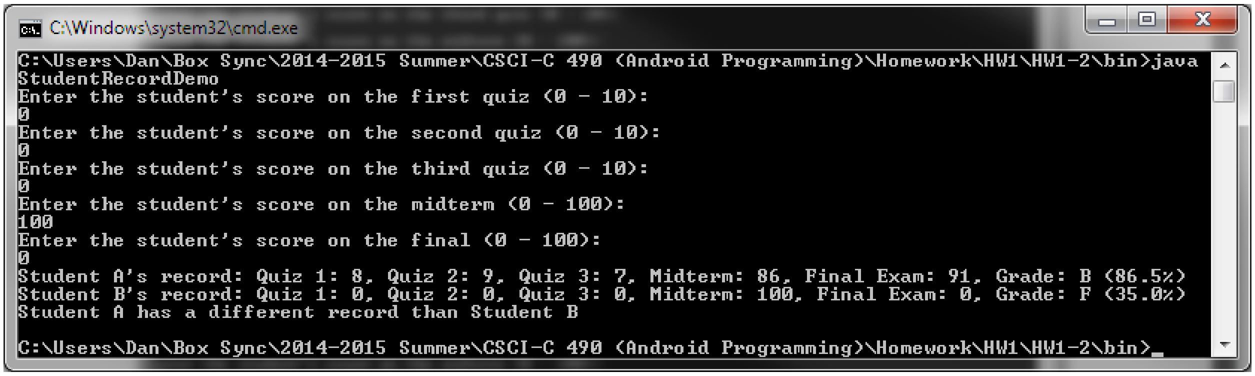
```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-2\bin>java
StudentRecordDemo
Enter the student's score on the first quiz (0 - 10):
9
Enter the student's score on the second quiz (0 - 10):
8
Enter the student's score on the third quiz (0 - 10):
9
Enter the student's score on the midterm (0 - 100):
95
Enter the student's score on the final (0 - 100):
96
Student A's record: Quiz 1: 8, Quiz 2: 9, Quiz 3: 7, Midterm: 86, Final Exam: 91, Grade: B (86.5%)
Student B's record: Quiz 1: 9, Quiz 2: 8, Quiz 3: 9, Midterm: 95, Final Exam: 96, Grade: A (93.31666
666666666%)
Student A has a different record than Student B

C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-2\bin>_
```

Demonstration that the quiz weight is calculated properly.



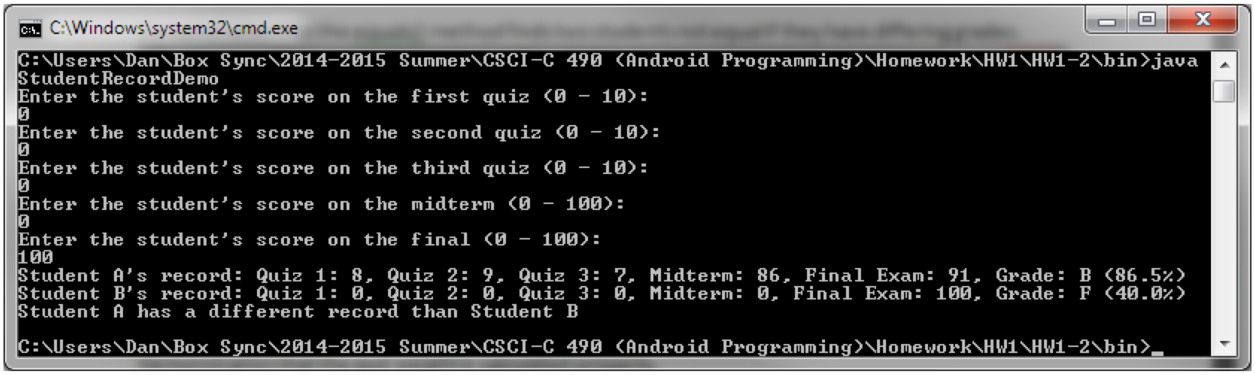
```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-2\bin>java
StudentRecordDemo
Enter the student's score on the first quiz (0 - 10):
10
Enter the student's score on the second quiz (0 - 10):
10
Enter the student's score on the third quiz (0 - 10):
10
Enter the student's score on the midterm (0 - 100):
0
Enter the student's score on the final (0 - 100):
0
Student A's record: Quiz 1: 8, Quiz 2: 9, Quiz 3: 7, Midterm: 86, Final Exam: 91, Grade: B (86.5%)
Student B's record: Quiz 1: 10, Quiz 2: 10, Quiz 3: 10, Midterm: 0, Final Exam: 0, Grade: F (25.0%)
Student A has a different record than Student B

C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-2\bin>_
```

Demonstration that the midterm weight is calculated properly.



```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-2\bin>java
StudentRecordDemo
Enter the student's score on the first quiz (0 - 10):
0
Enter the student's score on the second quiz (0 - 10):
0
Enter the student's score on the third quiz (0 - 10):
0
Enter the student's score on the midterm (0 - 100):
100
Enter the student's score on the final (0 - 100):
0
Student A's record: Quiz 1: 8, Quiz 2: 9, Quiz 3: 7, Midterm: 86, Final Exam: 91, Grade: B (86.5%)
Student B's record: Quiz 1: 0, Quiz 2: 0, Quiz 3: 0, Midterm: 100, Final Exam: 0, Grade: F (35.0%)
Student A has a different record than Student B

C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-2\bin>_
```

Demonstration that the final exam weight is calculated properly.



```
C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-2\bin>java
StudentRecordDemo
Enter the student's score on the first quiz (0 - 10):
0
Enter the student's score on the second quiz (0 - 10):
0
Enter the student's score on the third quiz (0 - 10):
0
Enter the student's score on the midterm (0 - 100):
0
Enter the student's score on the final (0 - 100):
100
Student A's record: Quiz 1: 8, Quiz 2: 9, Quiz 3: 7, Midterm: 86, Final Exam: 91, Grade: B (86.5%)
Student B's record: Quiz 1: 0, Quiz 2: 0, Quiz 3: 0, Midterm: 0, Final Exam: 100, Grade: F (40.0%)
Student A has a different record than Student B

C:\Users\Dan\Box Sync\2014-2015 Summer\CSCI-C 490 (Android Programming)\Homework\HW1\HW1-2\bin>_
```