

ASDV 1220, Programming Fundamentals

MP3

1. Create and implement class **FindHighestScore**, a program that prompts the user to enter the number of students and each student's name and score, and finally displays the name of the student with the highest score.

```
1 package loops;
2 import java.util.Scanner;
3
4 public class FindHighestScore
5 { public static void main(String[] args)
6 {
7     Scanner input = new Scanner(System.in);
8     System.out.print( "Enter the number of students: ");
9     int numofStudents = input.nextInt();
10    System.out.print("Enter a student name: ");
11    String student1 = input.next();
12    System.out.print("Enter a student score: ");
13    double score1 = input.nextDouble();
14    for (int i = 0; i < numofStudents - 1; i++){
15        System.out.print("Enter a student name: ");
16        String student = input.next();
17        System.out.print("Enter a student score: ");
18        double score = input.nextDouble();
19        if (score > score1){
20            student1 = student;
21            score1 = score;
22        }
23    }
24    System.out.println("Top student "+ student1 + "'s score is " + score1);
25 }
26 }
```

Output - Lab5 (run)

```
run:
Enter the number of students: 3
Enter a student name: John
Enter a student score: 88
Enter a student name: Mary
Enter a student score: 92
Enter a student name: Paul
Enter a student score: 52
Top student Mary's score is 92.0
BUILD SUCCESSFUL (total time: 33 seconds)
```

2. Create and implement class **FindHighestScoreWhile** by replacing the FOR loop of problem 1 with a WHILE loop.
3. Create and implement class **FindHighestScoreDoWhile** by replacing the FOR loop of problem 1 with a DO-WHILE loop.

4. Create and implement class **FindTwoHighestScores**, write a program that prompts the user to enter the number of students and each student's name and score, and finally displays the student with the highest score and the student with the second-highest score. USE a **FOR** loop.

```
1 package loops;
2
3 import ...
4
5 public class FindTwoHighestScores
6 {
7     public static void main(String[] args)
8     {...61 lines }
9 }
69
70
```

Output - Lab5 (run)

```
run:
Enter the number of students: 4
Enter a student name: john
Enter a student score: 100
Enter a student name: mary
Enter a student score: 90
Enter a student name: paul
Enter a student score: 95
Enter a student name: anna
Enter a student score: 99
Top two students:
john's score is 100.0
anna's score is 99.0
BUILD SUCCESSFUL (total time: 49 seconds)
```

5. Create and implement class **FindTwoHighestScoresWhile** by replacing the FOR loop of problem 4 with a WHILE loop.
6. Create and implement class **FindTwoHighestScoresDoWhile** by replacing the FOR loop of problem 4 with a DO-WHILE loop.

7. Create and implement class **GCD**(Greatest Common Divisor) Write a program that prompts the user to enter two positive integers and displays the gcd. Understand how the program below WORKS by tracing the code and USING the Debugger.

```
1 package lab5;
2 public class GCD
3 {
4     public static void main(String[] args)
5     {
6         java.util.Scanner input = new java.util.Scanner(System.in);
7         System.out.print("Enter the first number: ");
8         int n1 = input.nextInt();
9         System.out.print("Enter the second number: ");
10        int n2 = input.nextInt();
11
12        int d = (n1 < n2) ? n1 : n2;
13        for (; d >= 1; d--)
14            if ((n1 % d == 0) && (n2 % d == 0))
15                break;
16        System.out.println("GCD of " + n1 + " and " + n2 + " is " + d);
17    }
18 }
19
```

Output

```
run:
Enter the first number: 20
Enter the second number: 8
GCD of 20 and 8 is 4
BUILD SUCCESSFUL (total time: 30 seconds)
```

8. Create and implement class **GCDWhile** by replacing the FOR loop of problem 7 with a WHILE loop. You have to use break.
9. Create and implement class **GCDDoWhile** by replacing the FOR loop of problem 7 with a DO-WHILE loop. You have to use break.

10. Create and implement Calendar as given below. The program that prompts the user to enter the year and first day of the year and displays the calendar table for the year on the console. For example, if the user entered the year 2013, and 2 for Tuesday, January 1, 2013, your program should display the calendar for each month in the year, as follows:

```

Output
Debugger C

December 2013
-----
Sun Mon Tue Wed Thu Fri Sat
  1  2  3  4  5  6  7
  8  9 10 11 12 13 14
 15 16 17 18 19 20 21
 22 23 24 25 26 27 28
 29 30 31

BUILD SUCCESSFUL (total time: 41 seconds)

```

```

1 package loops;
2 import java.util.Scanner;
3 public class Calendar
4 {
5     public static void main(String[] args)
6     { Scanner input = new Scanner(System.in);
7       System.out.print("Enter a year: ");
8       int year = input.nextInt();
9       System.out.print("Enter the first day of the year: ");
10      int firstDay = input.nextInt();
11
12      int startDay = firstDay;
13      int numberOfDaysInMonth = 0;
14      for (int month = 1; month <= 12; month++){
15          System.out.print(" ");
16          switch (month){
17              case 1: System.out.println("January " + year); numberOfDaysInMonth = 31; break;
18              case 2: System.out.println("February " + year);
19                  if (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0)) numberOfDaysInMonth = 29;
20                  else numberOfDaysInMonth = 28;
21                  break;
22              case 3: System.out.println("March " + year); numberOfDaysInMonth = 31; break;
23              case 4: System.out.println("April " + year); numberOfDaysInMonth = 30; break;
24              case 5: System.out.println("May " + year); numberOfDaysInMonth = 31; break;
25              case 6: System.out.println("June " + year); numberOfDaysInMonth = 30; break;
26              case 7: System.out.println("July " + year); numberOfDaysInMonth = 31; break;
27              case 8: System.out.println("August " + year); numberOfDaysInMonth = 31; break;
28              case 9: System.out.println("September " + year); numberOfDaysInMonth = 30; break;
29              case 10: System.out.println("October " + year); numberOfDaysInMonth = 31; break;
30              case 11: System.out.println("November " + year); numberOfDaysInMonth = 30; break;
31              case 12: System.out.println("December " + year); numberOfDaysInMonth = 31; break;
32          }
33          System.out.println("-----");
34          System.out.println(" Sun Mon Tue Wed Thu Fri Sat");
35          int i = 0;
36          for (i = 0; i < startDay; i++)
37              System.out.print(" ");
38          for (i = 1; i <= numberOfDaysInMonth; i++){
39              if (i < 10) System.out.print(" " + i);
40              else System.out.print("  " + i);
41              if ((i + startDay) % 7 == 0) System.out.println();
42          }
43          System.out.println();
44          System.out.println();
45          startDay = (startDay + numberOfDaysInMonth) % 7;
46      }
47  }
48 }
49
50

```

11. Create and implement class **CalendarWhile** by:
- Replacing the FOR loop of problem 10 with a WHILE loop.
 - Replacing the SWITCH with IFs.

12. (9 points) Create and implement class **Stats** to compute mean and standard deviation of a set of numbers entered from the use.) In business applications, you are often asked to compute the mean and standard deviation of data. The mean is simply the average of the numbers. The standard deviation is a statistic that tells you how tightly all the various data are clustered around the mean in a set of data. For example, what is the average age of the students in a class? How close are the ages? If all the students are the same age, the deviation is 0.

Your program behaves as show below:

Displays a menu to the user to enter Q/q to quit, or Y/y to start entering numbers.

If the user types Y/y prompt the user to start entering numbers or -1 to stop entering numbers.

When the user stops entering numbers displays the mean and standard deviation of these numbers using the following formula:

$$\text{mean} = \frac{\sum_{i=1}^n x_i}{n} = \frac{x_1 + x_2 + \dots + x_n}{n} \quad \text{deviation} = \sqrt{\frac{\sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i\right)^2}{n}}{n - 1}}$$

Special cases to handle:

- Handle the case the user enters invalid input instead of Q/q, Y/y as shown below.
- Handle the case that the user selects Y/y and then immediately types -1(i.e. No numbers entered).
- Handle the case that the numbers entered is just one number . Then the standard deviation is zero.

```
3 public class Stats
4 {public static void main(String[] args)
5 [+ {...54 lines }
```

Output

Debugger Console

Lab5 (run)

run:

```
Q/q: to quit or Y/y to start a new series of STATS:lafayette
      ????: Invalid character!
Q/q: to quit or Y/y to start a new series of STATS:y
      ====> Enter a number or -1 to stop entering numbers: -1
      ????: You entered no numbers!
Q/q: to quit or Y/y to start a new series of STATS:y
      ====> Enter a number or -1 to stop entering numbers: 33
      ====> Enter a number or -1 to stop entering numbers: -1
+++=> The mean is 33.0
+++=> The standard deviation is 0.0

Q/q: to quit or Y/y to start a new series of STATS:y
      ====> Enter a number or -1 to stop entering numbers: 2
      ====> Enter a number or -1 to stop entering numbers: 3.7
      ====> Enter a number or -1 to stop entering numbers: 11
      ====> Enter a number or -1 to stop entering numbers: -1
+++=> The mean is 5.566666666666666
+++=> The standard deviation is 4.781561809004809

Q/q: to quit or Y/y to start a new series of STATS:q
+++=> Goodbye!
BUILD SUCCESSFUL (total time: 1 minute 21 seconds)
```