*LAB # 03*

ARRAYS

# *OBJECTIVE:*

*To Study Java one dimensional and two dimensional arrays*

LAB TASK

**Task # 01:**

*Write a program that reads (fictitious) student test scores in the range 0 through 100 and print the following statistics to two decimal places:*

*The average (mean) score.*

*The student with the highest score.*

*The student with the lowest score.*

*The number of students whose score equal or exceed the average.*

*For each student:*

*The difference between the average score and the student’s score (this can be either positive or negative).*

*The grade letter where*

*A is a score of 90 or greater.*

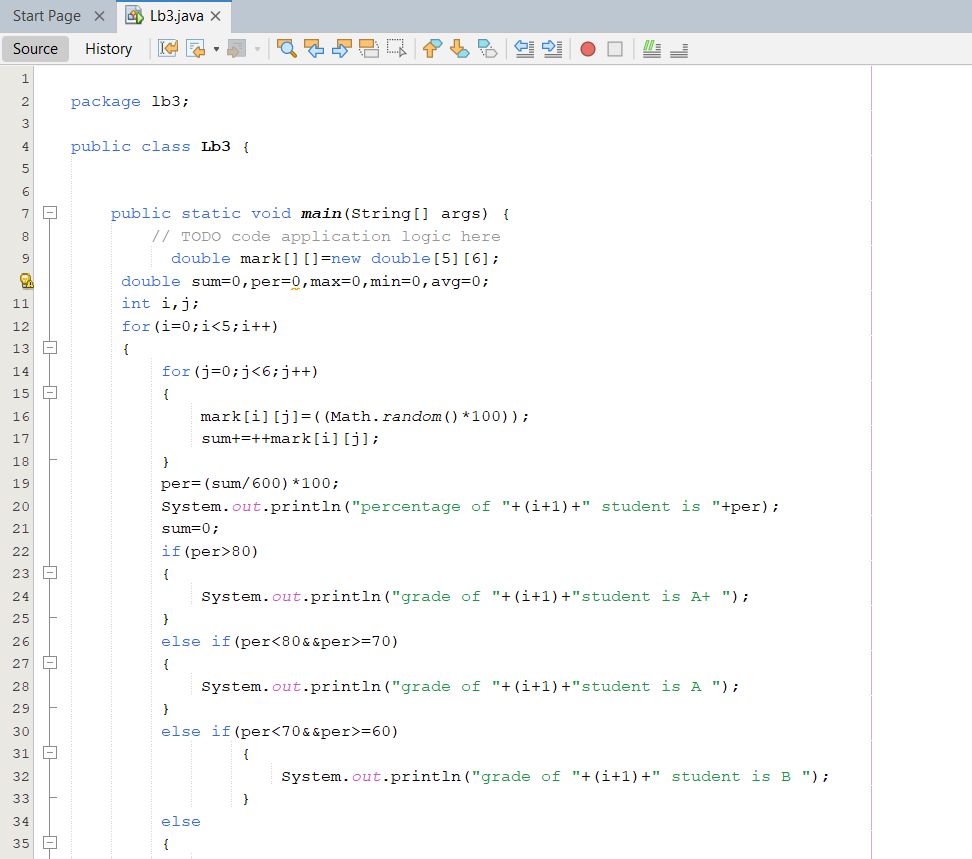
*B is a score of 80 through 89.99.*

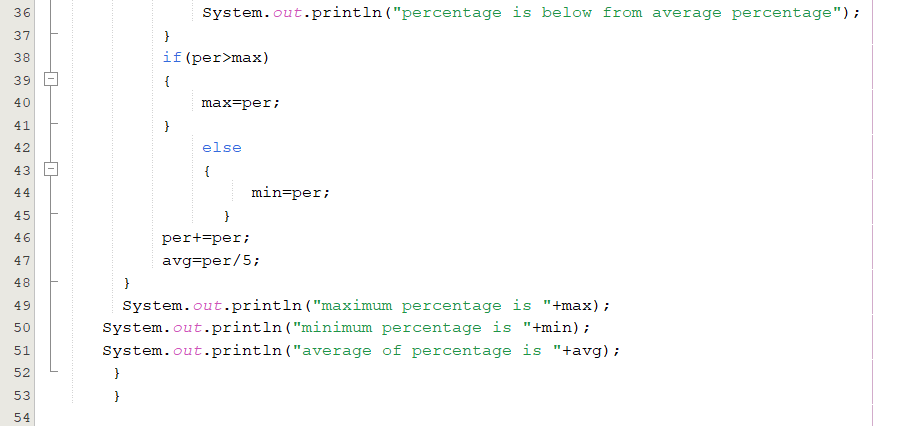
*C is a score of 70 through 79.99*

*D is a score of 60 through 69.99*

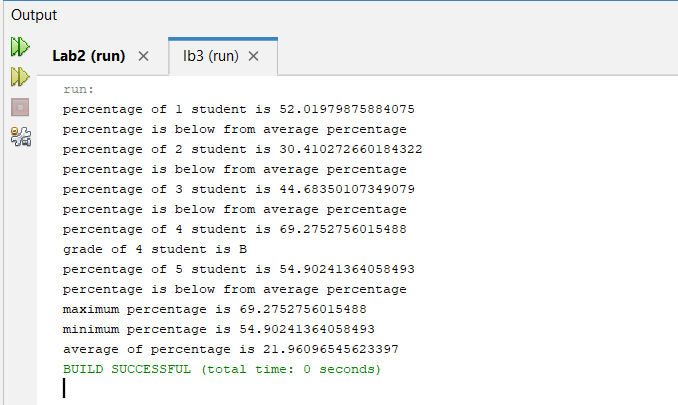
*E is a score of less than 60.*

**Code:**





**Output**



**Task # 02:**

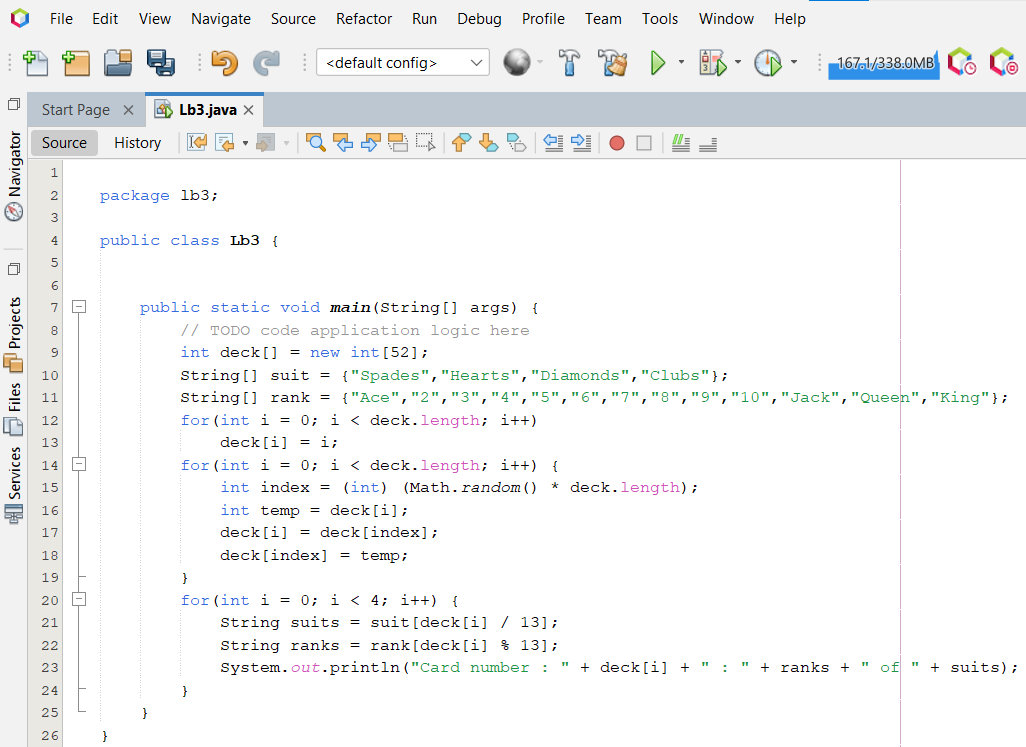
*The problem is to write a program that picks four cards randomly from a deck of 52 cards. All the cards can be represented using an array named deck, filled with initial values 0 to 51, as follows:*

*Int [ ] deck = new int[52];*

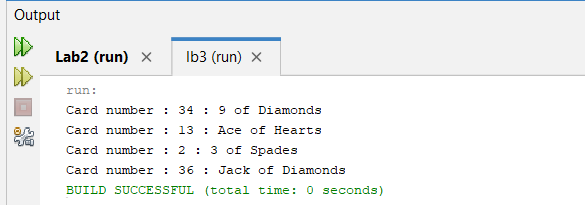
*// Initialize cards for (int i = 0; i < deck.length; i++) deck[i] = i;*

*Card numbers 0 to 12, 13 to 25, 26 to 38, 39 to 51 represent 13 Spades, 13 Hearts, 13 Diamonds, and 13 Clubs, respectively, as shown in Figure 6.3. After shuffling the array deck, pick the first four cards from deck*.

**Code:**

****

**Output:**

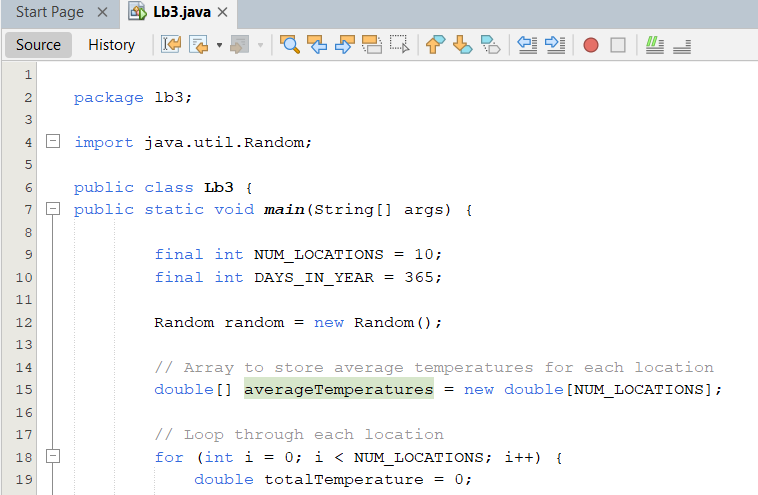


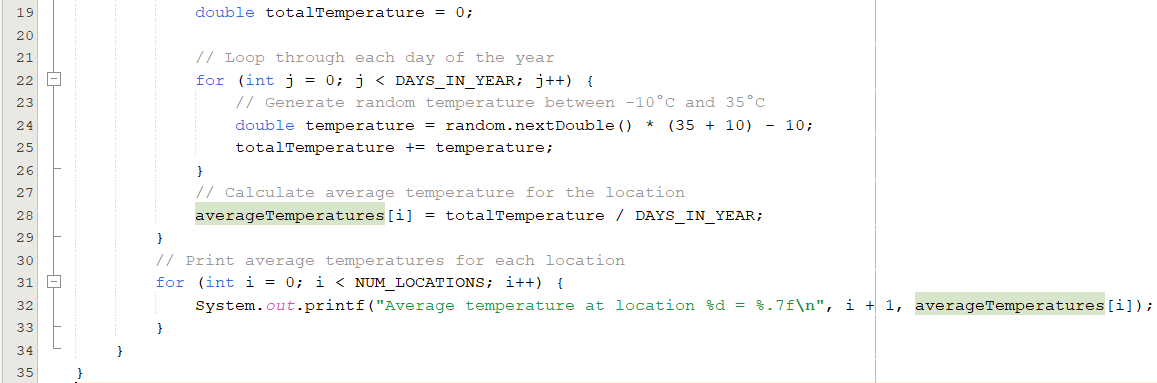
**Task # 03:**

*Write a program to wander around 10 different locations and find their average temperature of a year, you’ll generate the temperatures as random values between -10 degrees and 35 degrees. This assumes you are recording temperatures in degrees Celsius. If you prefer Fahrenheit, you could generate values from 14 degrees to 95 degrees to cover the same range.*

*Expected Output: Average temperature at location 1 = 12.2733345 Average temperature at location 2 = 12.012519 Average temperature at location 3 = 11.545245…continue till location 10*

**Code:**

****

****

**Output:**

