**Programming Paradigms Laboratory**

**B.Tech.**



**Name : Subhendu Maji**

**Roll Number : 18ETCS002121**

**Department : Computer Science and Engineering**

**Faculty of Engineering & Technology**

**Ramaiah University of Applied Sciences**

|  |  |
| --- | --- |
| Faculty | Engineering & Technology |
| Programme | B. Tech. in Computer Science and Engineering |
| Year/Semester | 2nd Year / 4th Semester |
| Name of the Laboratory | Programming Paradigms Laboratory |
| Laboratory Code | 19CSL217A |

# Laboratory 3

Title of the Laboratory Exercise: One dimensional arrays

1. Questions
2. Develop a Java program to delete the duplicate elements from an array.
3. Develop a GradeBook class with an instance variables string course name and array of grades that instructors can use to maintain students’ grades on an exam and display a grade report that includes the grades, class average, lowest grade and highest grade.
4. Calculations/Computations/Algorithms

**Algorithm 1:**

Step 1: start

Step 2: input size of array , say n

Step 3: input n elements in array ,say arr

Step 4: for i = 0 to n:

4.1 for for j =i+1 to n

4.1.1 if arr[i] == arr[j]

4.1.1.1 arr[j] == arr[n-1]

4.1.1.2 n - -

4.1.1.3 j - -

Step 5: print arr

Step 6: stop

**Algorithm 2:**

Step 1: start

Step 2: create a class grade

Step 3: initialize instance variables String course and Array grades

Step 4: create a method GradeReport

4.1 for i : grade

4.1.1 if 0 < = i < 10 : report[1] += ‘\*’

4.1.2 if 10 < = i < 20 : report[2] += ‘\*’

4.1.3 if 20 < = i < 30 : report[3] += ‘\*’

4.1.4 if 30 < = i < 40 : report[4] += ‘\*’

. . .

4.1.10 if i == 100 : report[10] += ‘\*’

Step 5: print

5.1 for j =1 to report.length

5.1.1 print (10 \* (j - 1)) + "-" + (10 \* j - 1) + " : " + report[j]

Step 6: create a class GradeMain

Step 7: create grades object with the help of constructor

Step 8: call GradeReport()

Step 9: stop

1. Presentation of Results



Figure delete duplicate elements from an array

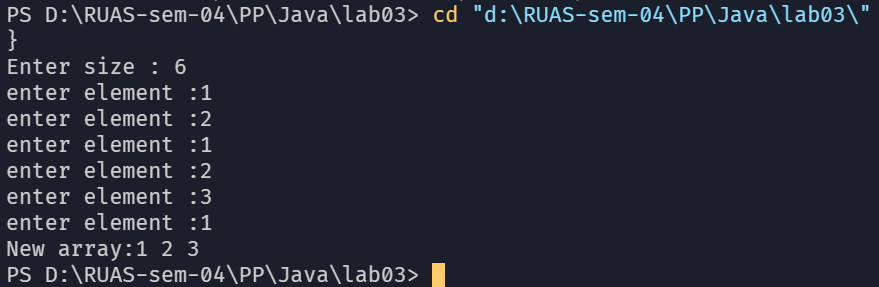


Figure output of deleting duplicate elements

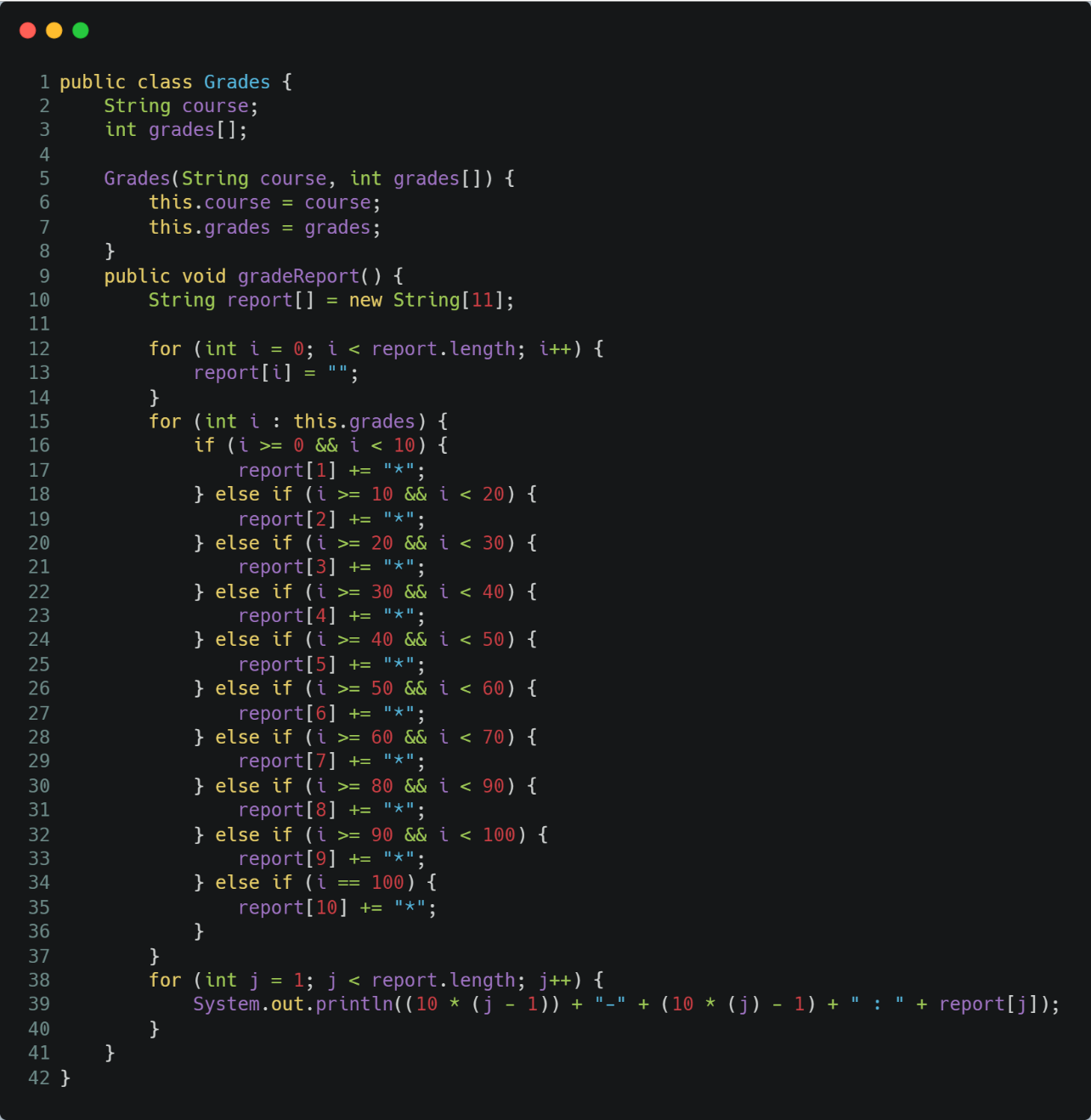


Figure Grades class

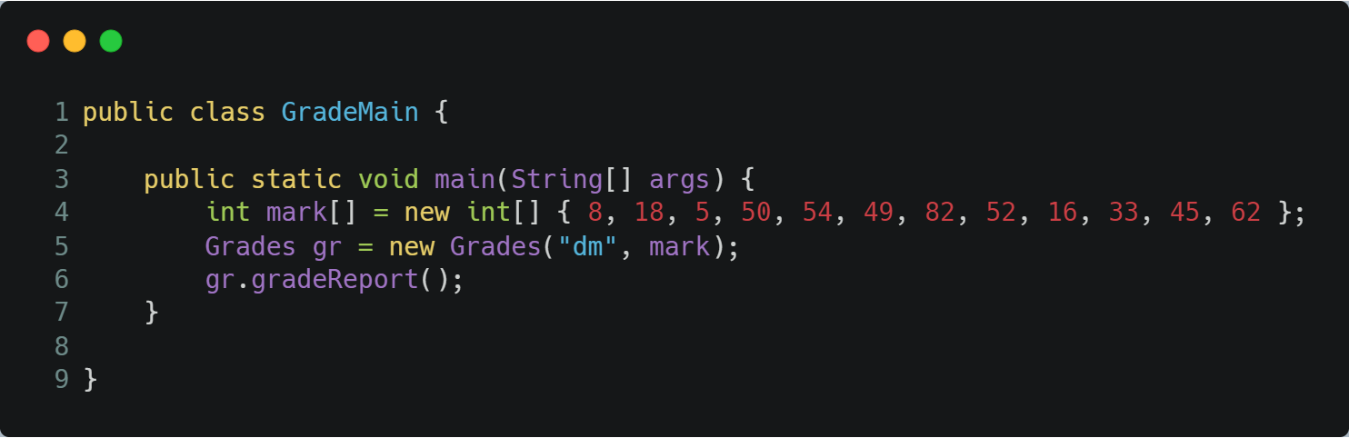


Figure GradeMain Class

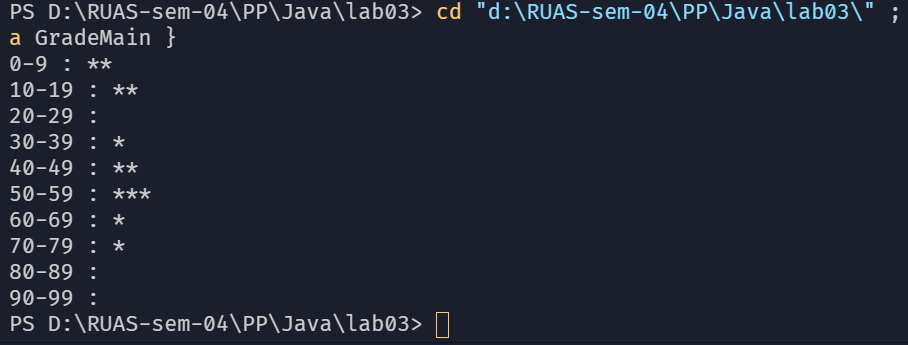


Figure output of GradeReport

1. Conclusions

Java String class provides a lot of methods to perform operations on strings such as **compare**(), **concat**(), **equals**(), **split**(), **length**(), **replace**(), **compareTo**(), **intern**(), **substring**() etc.

The Java String is immutable which means it cannot be changed. Whenever we change any string, a new instance is created. For mutable strings, we can use **StringBuffer** and **StringBuilder** classes.