**LAB # 01**

**INTRODUCTION TO STRING POOL, LITERALS, AND WRAPPER CLASSES**

LAB TASKS

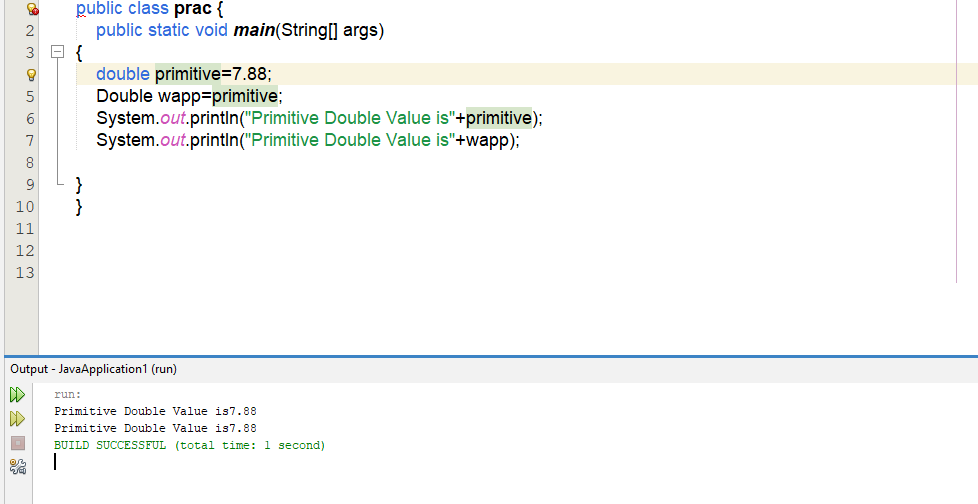
1. Write a program that initialize five different strings using all the above mentioned ways, i.e., a) string literals b) new keyword also use intern method and show string immutability.

**Code:**

****

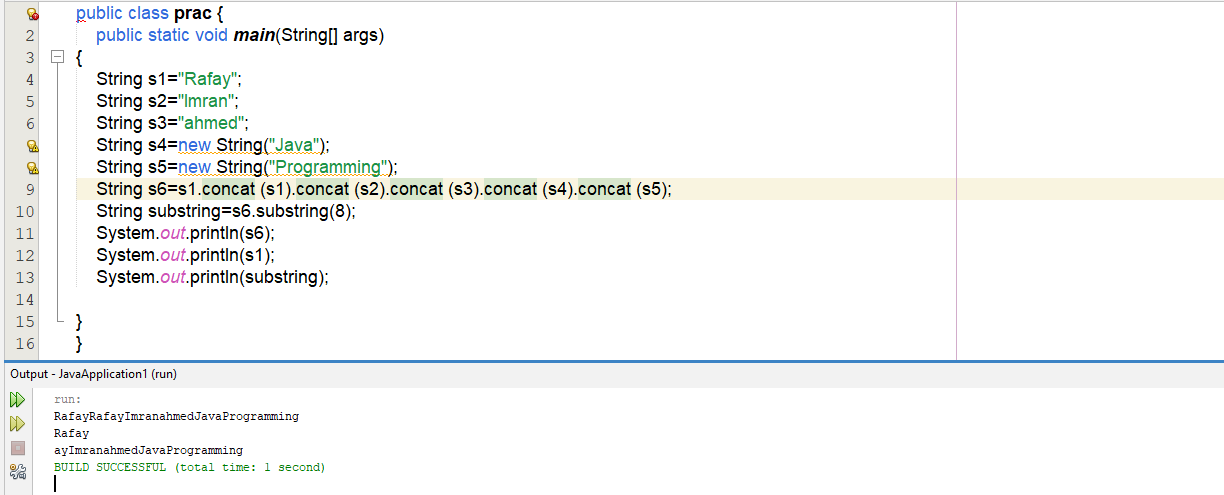
2. Write a program to convert primitive data type Double into its respective wrapper object.

**Code:**

****

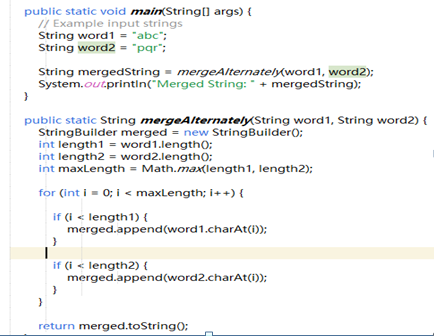
3. Write a program that initialize five different strings and perform the following operations. a. Concatenate all five stings. b. Convert fourth string to uppercase. c. Find the substring from the concatenated string from 8 to onward

**Code:**

****

4.You are given two strings word1 and word2. Merge the strings by adding letters in alternating order, starting with word1. If a string is longer than the other, append the additional letters onto the end of the merged string. Return the merged string. Example: Input: word1 = "abc", word2 = "pqr" Output: "apbqcr" Explanation: The merged string will be merged as so: word1: a b c word2: p q r merged: a p b q c r

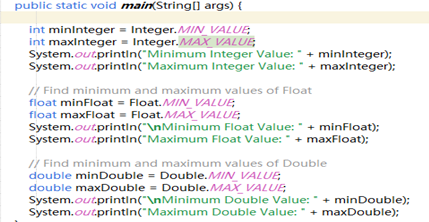
**Code:**

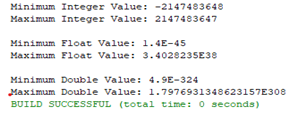




5. Write a Java program to find the minimum and maximum values of Integer, Float, and Double using the respective wrapper class constants.

**Code:**

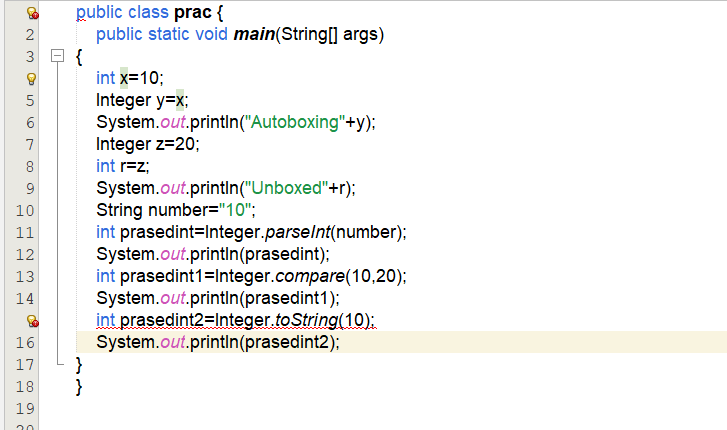


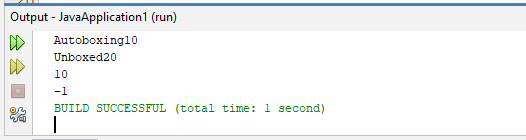


HOMETASKS

1. Write a JAVA program to perform Autoboxing and also implement different methods of wrapper class.

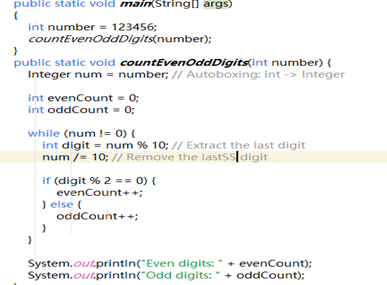
**Code:**

****



2. Write a Java program to count the number of even and odd digits in a given integer using Autoboxing and Unboxing.

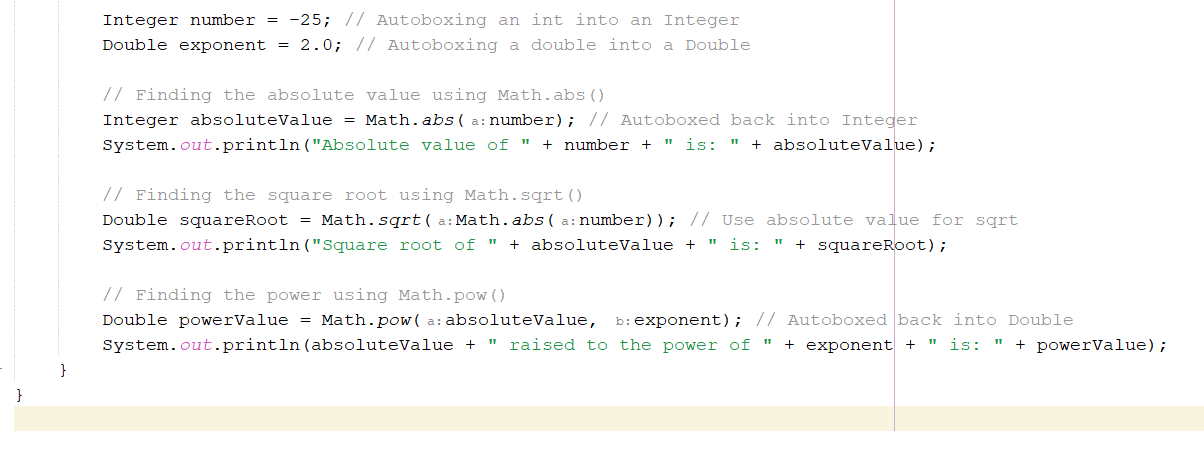
**Code:**

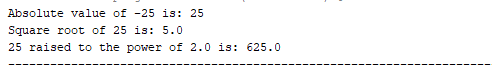




3. Write a Java program to find the absolute value, square root, and power of a number using Math class methods, while utilizing Autoboxing and Wrapper classes.

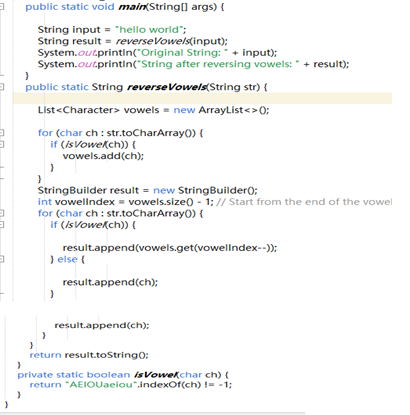
**Code:**

****

****

4. Write a Java program to reverse only the vowels in a string.

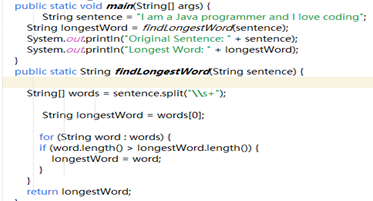
**Code:**

****

****

5. Write a Java program to find the longest word in a sentence.

**Code:**

****

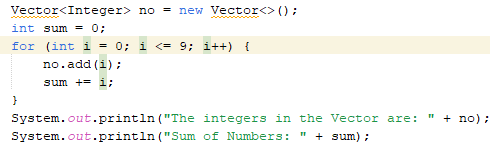
****

**LAB # 02**

**ARRAYLIST AND VECTOR IN JAVA**

LAB TASKS

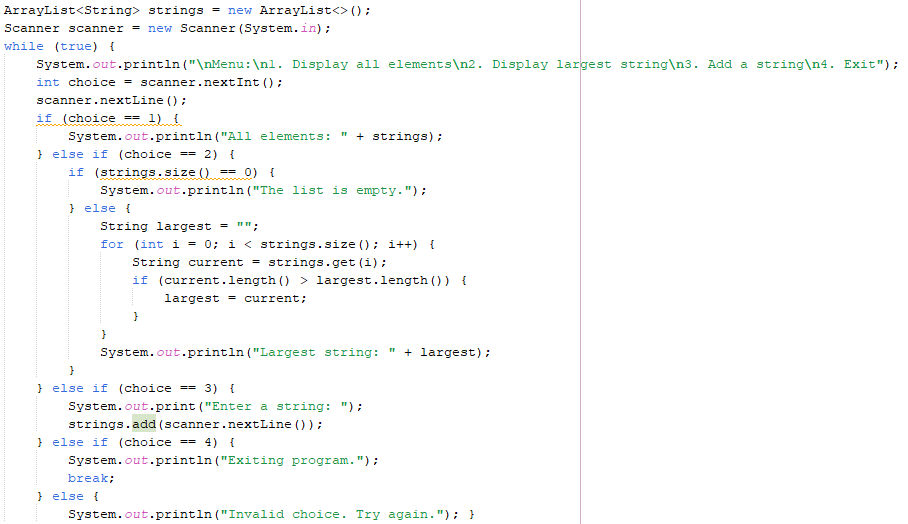
1. Write a program that initializes Vector with 10 integers in it. Display all the integers and sum of these integers.

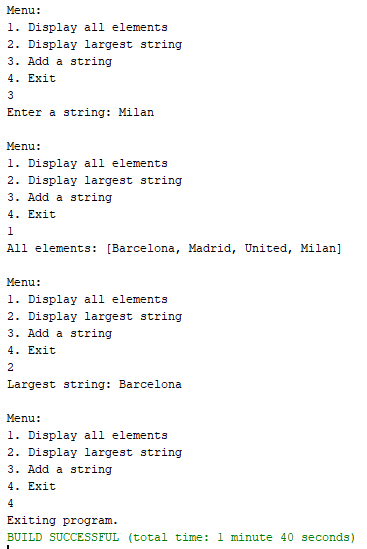
**Code:**

**Output:**

****

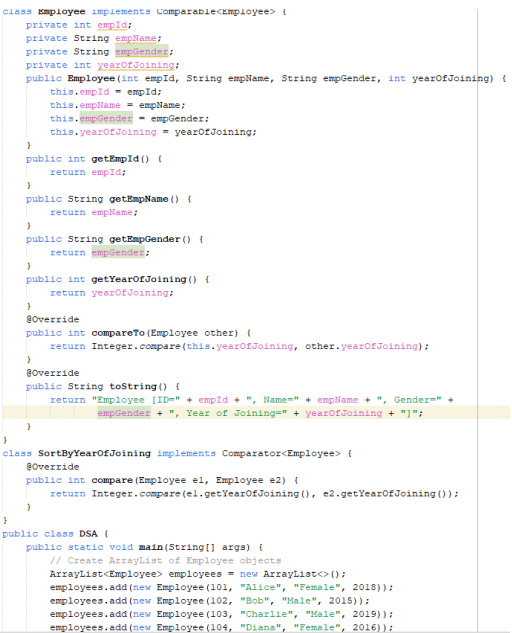
2. Create a ArrayList of string. Write a menu driven program which: a. Displays all the elements b. Displays the largest String



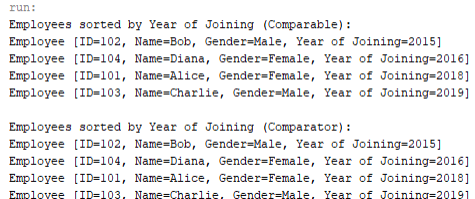
**Output:**

3. Create a Arraylist storing Employee details including Emp\_id, Emp\_Name, Emp\_gender, Year\_of\_Joining (you can also add more attributes including these). Then sort the employees according to their joining year using Comparator and Comparable interfaces.

**Code:**



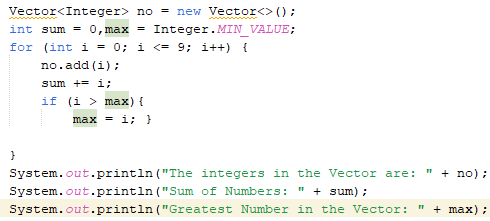
**Output:**



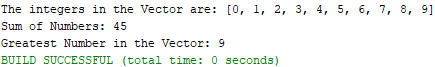
4. Write a program that initializes Vector with

10 integers in it. • Display all the integers • Sum of these integers. • Find Maximum Element in Vector

**Code:**

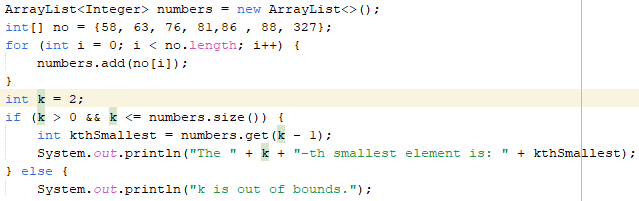


**Output:**



5. Find the k-th smallest element in a sorted ArrayList

**Code:**

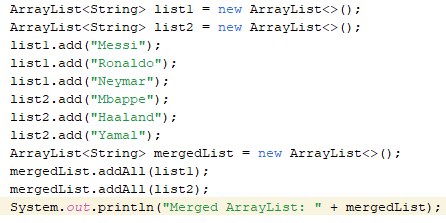


**Output:**



6. Write a program to merge two ArrayLists into one.

**Code:**

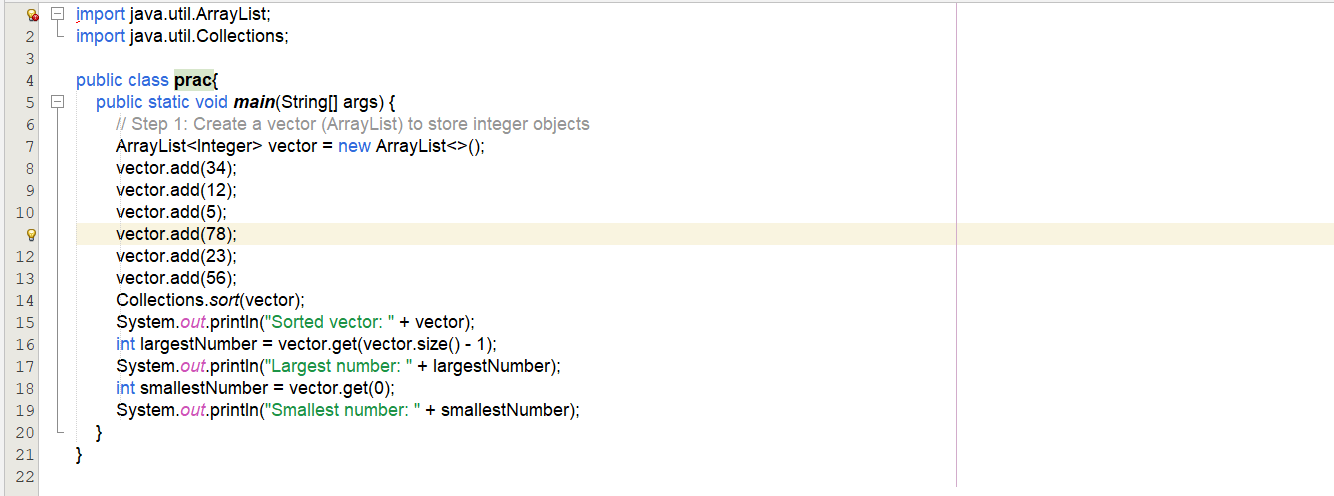


**Output:**



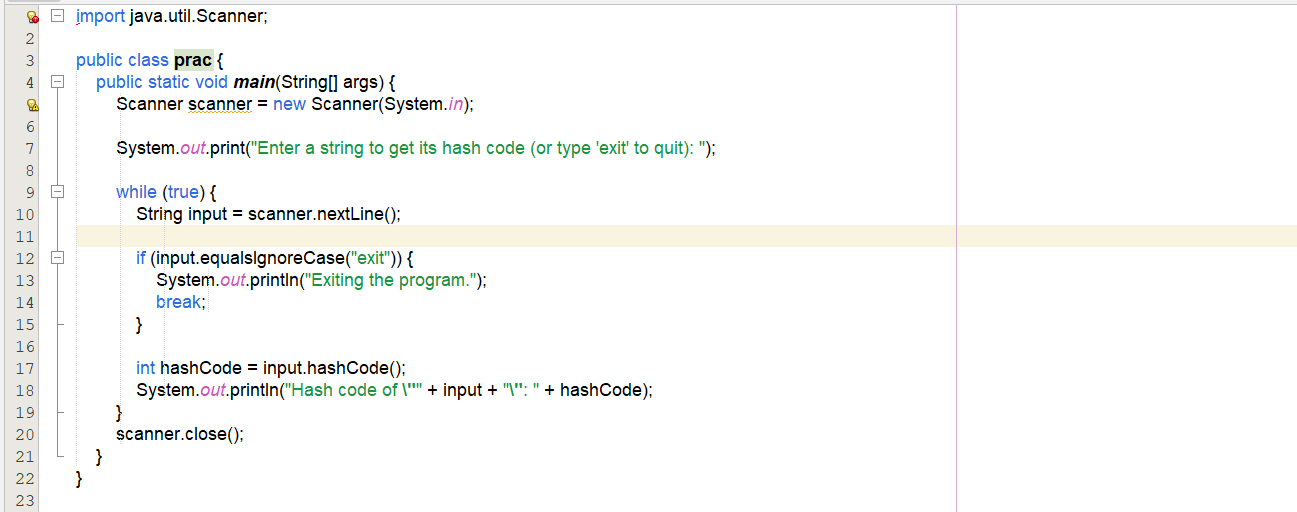
Home Tasks

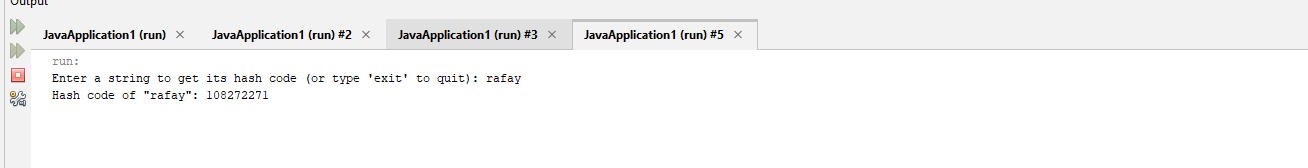
1. Create a Vector storing integer objects as an input. a. Sort the vector b. Display largest number c. Display smallest number





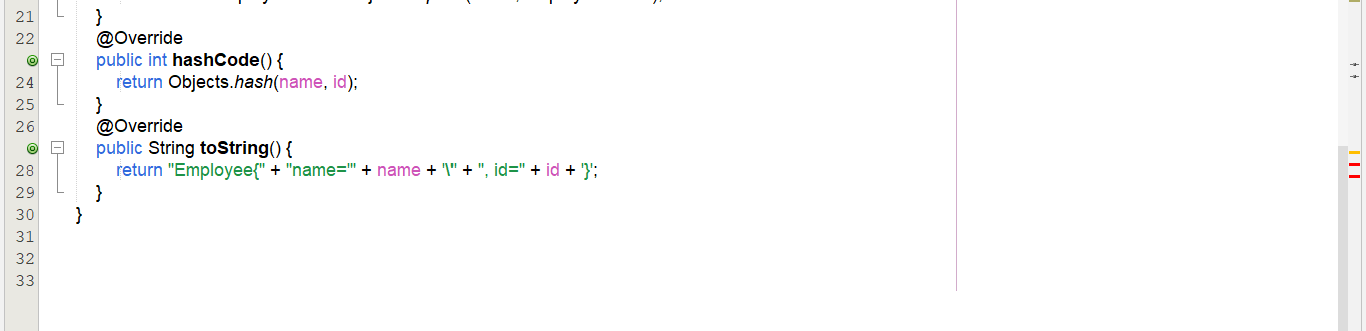
1. Write a java program which takes user input and gives hashcode value of those inputs using hashCode () method.

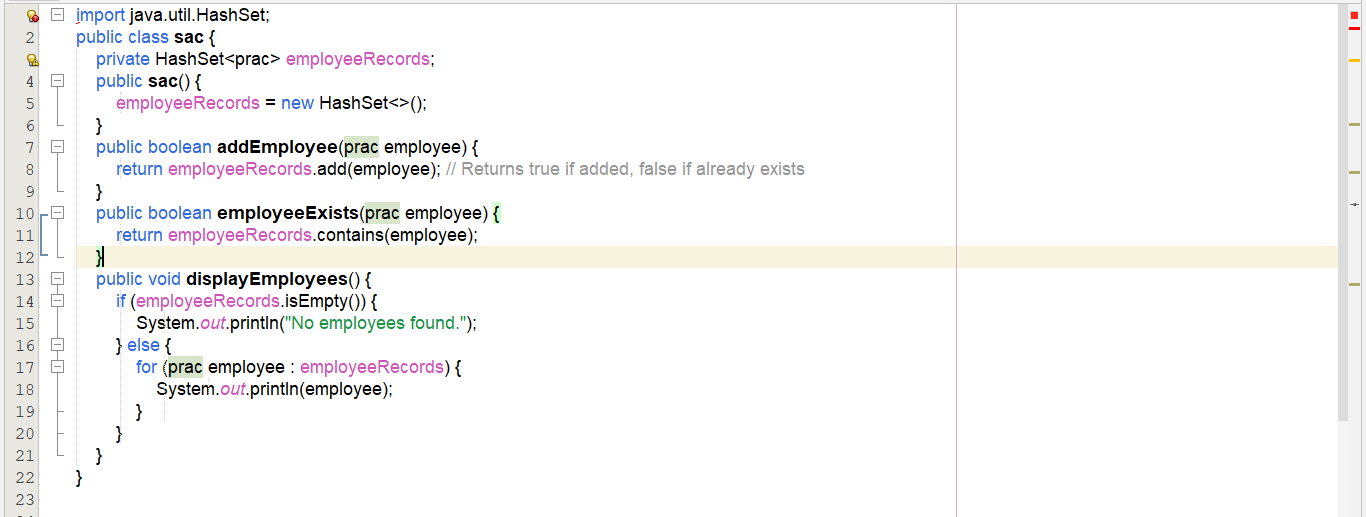


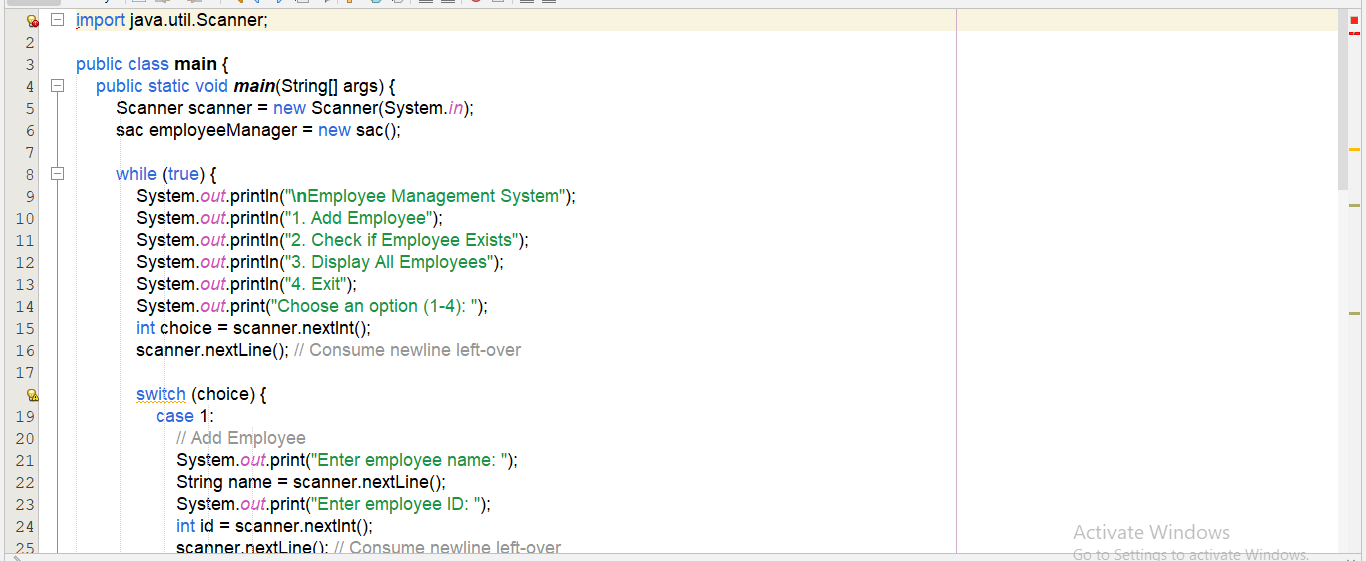


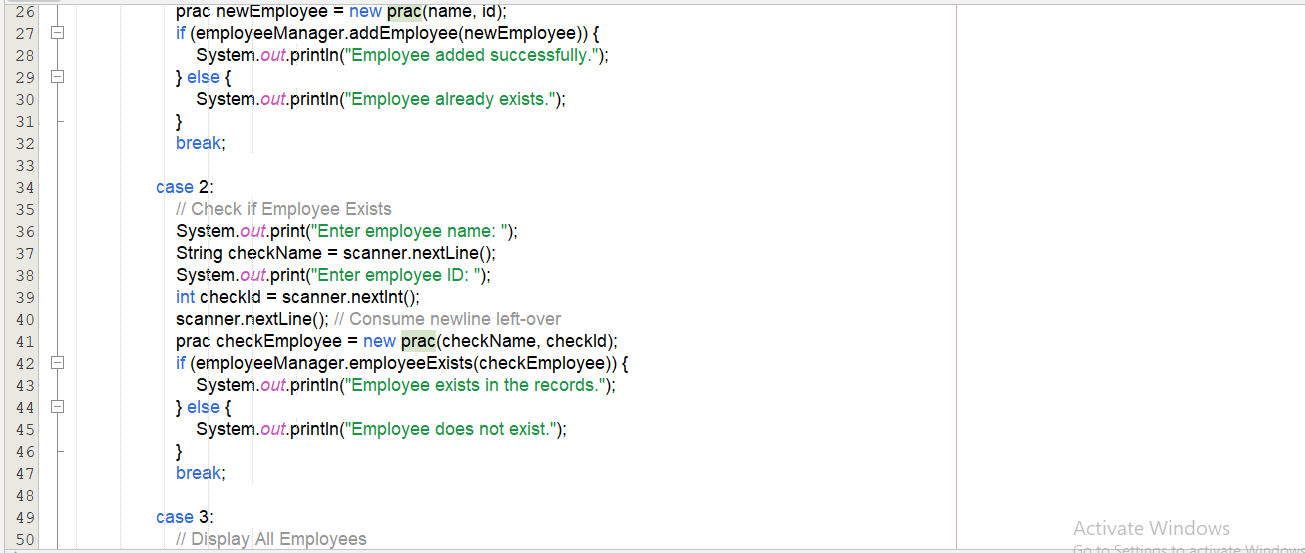
Create a java project, suppose you work for a company that needs to manage a list of employees. Each employee has a unique combination of a name and an ID. Your goal is to ensure that you can track employees effectively and avoid duplicate entries in your system. Requirements a. Employee Class: You need to create an Employee class that includes: • name: The employee's name (String). • id: The employee's unique identifier (int). • Override the hashCode() and equals() methods to ensure that two employees are considered equal if they have the same name and id. b. Employee Management: You will use a HashSet to store employee records. This will help you avoid duplicate entries. c. Operations: Implement operations to: • Add new employees to the record. • Check if an employee already exists in the records. • Display all employees.

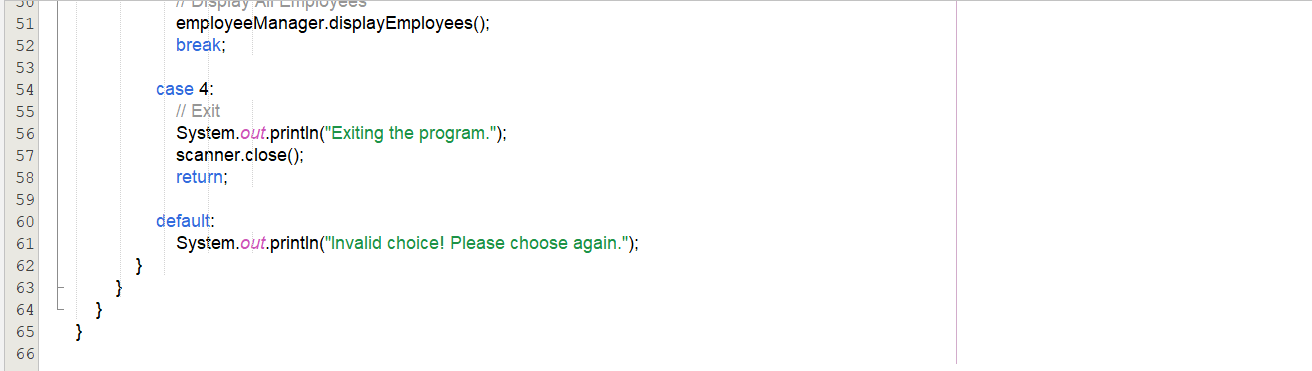


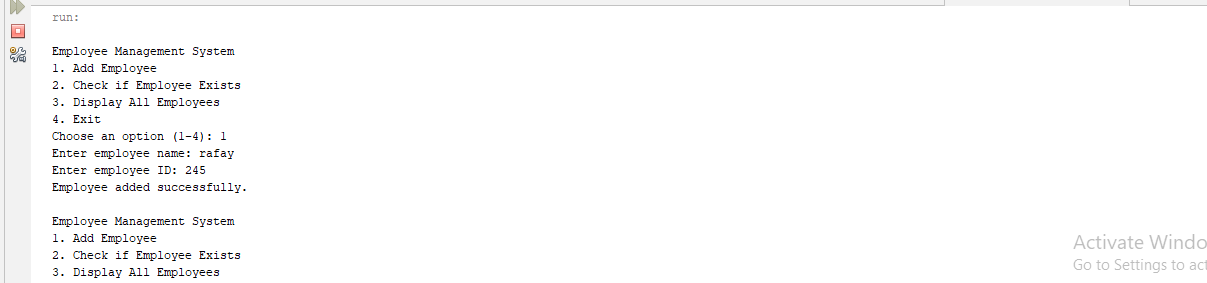












4.Create a Color class that has red, green, and blue values. Two colors are considered equal if their RGB values are the same

