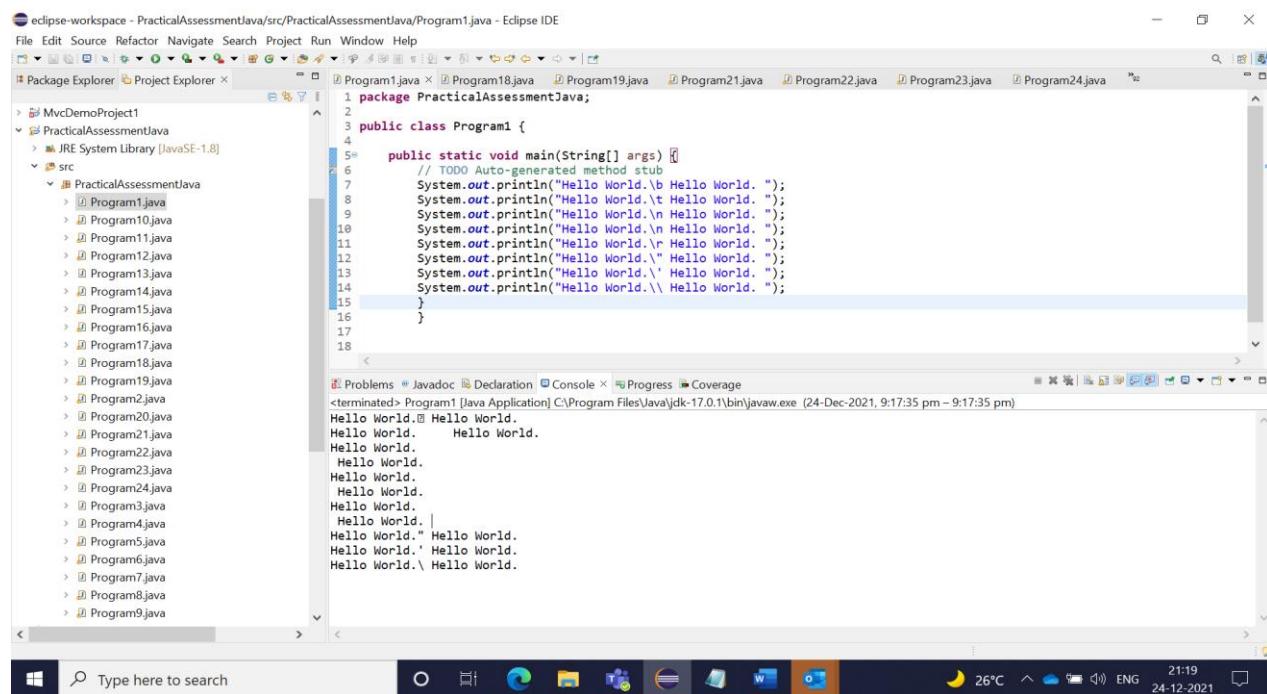


Name: Manjula B
Employee Id: 5770

Practical Assessment-Java Answers

1) Use the following escape sequence and print the statements given below and notice the output.

1. Hello World.\b Hello World.
2. Hello World.\t Hello World.
3. Hello World.\n Hello World.
4. Hello World.\n Hello World.
5. Hello World.\r Hello World.
6. Hello World.\\" Hello World.
7. Hello World.\' Hello World.
8. Hello World.\\ Hello World.



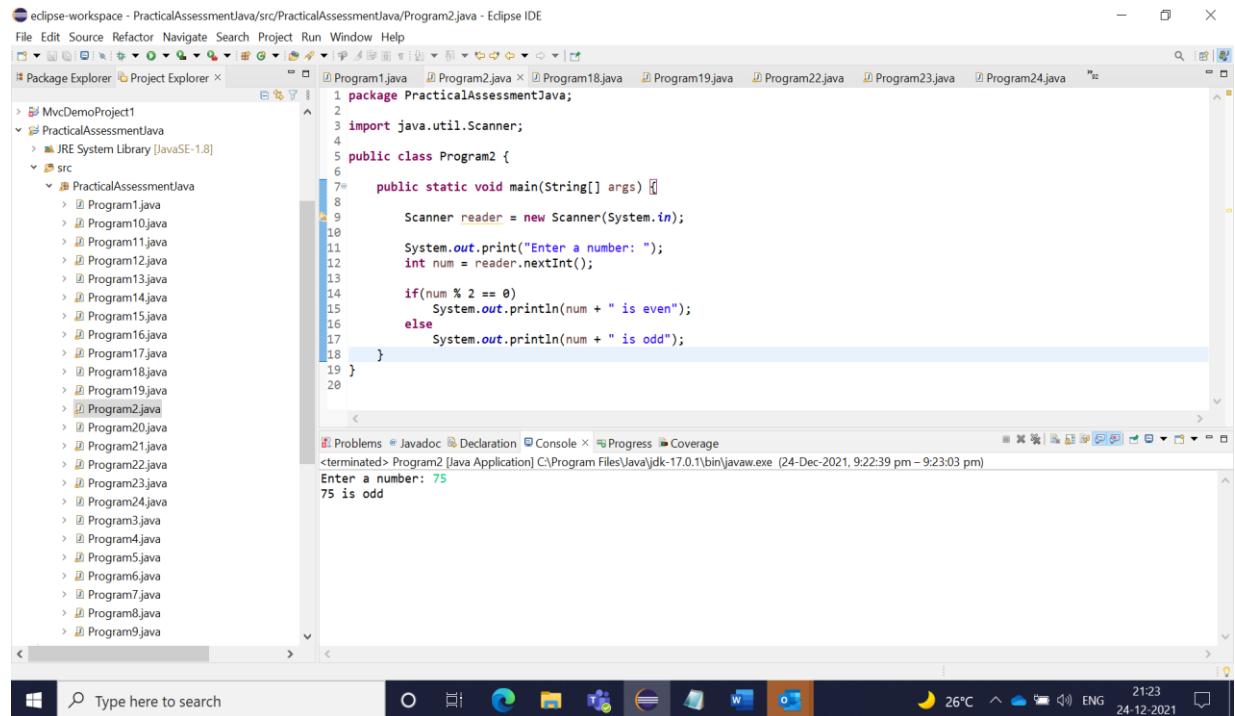
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with a package named "PracticalAssessmentJava" containing multiple Java files (Program1.java to Program24.java).
- Code Editor:** Displays the content of Program1.java, which contains a main method that prints various escape sequences to the console.
- Console Output:** Shows the terminal output of the Java application, demonstrating the printed escape sequences.
- System Tray:** At the bottom, it shows the date (24-12-2021), time (21:19), and system status (26°C, ENG).

```
1 package PracticalAssessmentJava;
2
3 public class Program1 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("Hello World.\b Hello World. ");
8         System.out.println("Hello World.\t Hello World. ");
9         System.out.println("Hello World.\n Hello World. ");
10        System.out.println("Hello World.\r Hello World. ");
11        System.out.println("Hello World.\\" Hello World. ");
12        System.out.println("Hello World.\' Hello World. ");
13        System.out.println("Hello World.\\ Hello World. ");
14        System.out.println("Hello World.\\\\ Hello World. ");
15    }
16}
17
18
```

```
2021-12-24 21:19:35,260:Hello World.\b Hello World. 
2021-12-24 21:19:35,260:Hello World.\t Hello World. 
2021-12-24 21:19:35,260:Hello World.\n Hello World. 
2021-12-24 21:19:35,260:Hello World.\r Hello World. 
2021-12-24 21:19:35,260:Hello World.\\" Hello World. 
2021-12-24 21:19:35,260:Hello World.\' Hello World. 
2021-12-24 21:19:35,260:Hello World.\\ Hello World. 
2021-12-24 21:19:35,260:Hello World.\\\\ Hello World. 
```

2) Accept a number from user and determine if a given number is odd or even and print



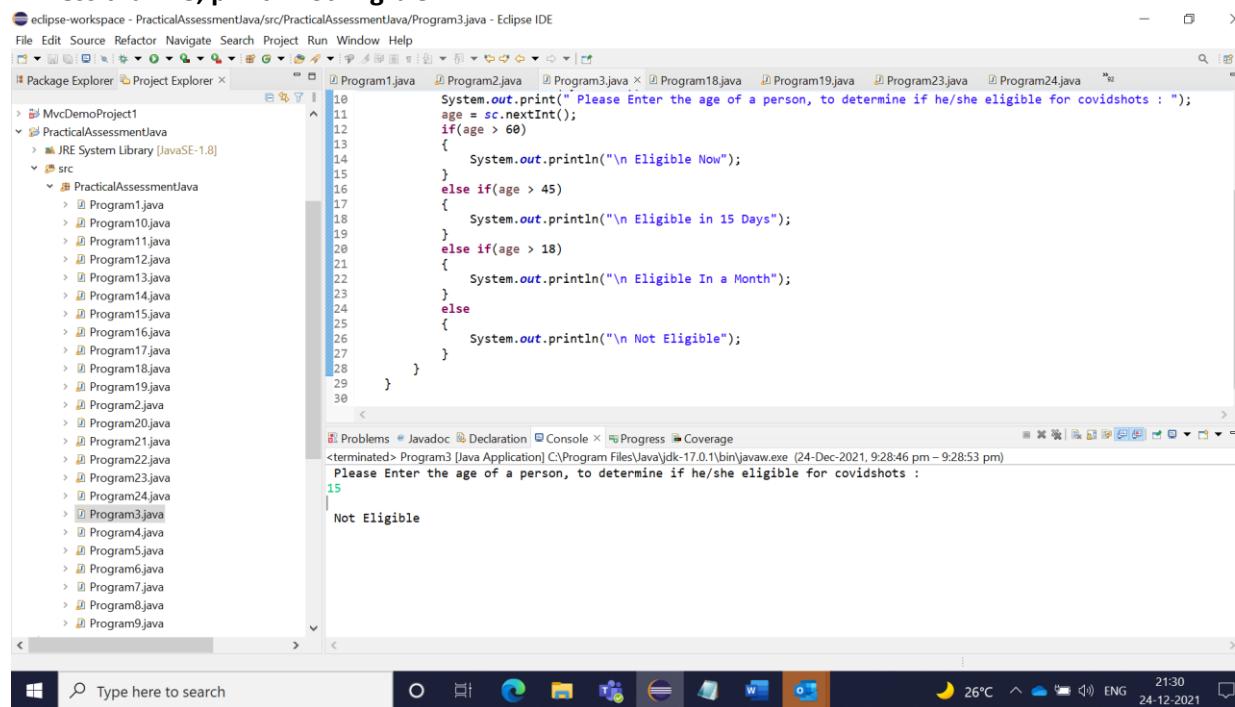
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "PracticalAssessmentJava" with a "src" folder containing multiple Java files (Program1.java, Program10.java, Program11.java, Program12.java, Program13.java, Program14.java, Program15.java, Program16.java, Program17.java, Program18.java, Program19.java, Program2.java, Program20.java, Program21.java, Program22.java, Program23.java, Program24.java).
- Code Editor:** Displays the content of Program2.java:

```
1 package PracticalAssessmentJava;
2
3 import java.util.Scanner;
4
5 public class Program2 {
6
7     public static void main(String[] args) {
8
9         Scanner reader = new Scanner(System.in);
10
11         System.out.print("Enter a number: ");
12         int num = reader.nextInt();
13
14         if(num % 2 == 0)
15             System.out.println(num + " is even");
16         else
17             System.out.println(num + " is odd");
18     }
19 }
20
```
- Console:** Shows the output of the program: "Enter a number: 75" and "75 is odd".
- System Tray:** Shows the Windows taskbar with the date and time (24-12-2021, 21:23).

3)Accept age of a person and determine if he/she eligible for covidshots.

1. More than 60, print "Eligible now"
2. More than 45, print "Eligible in 15 days"
3. More than 18, print "Eligible in a month"
4. Less than 18, print "Not Eligible"

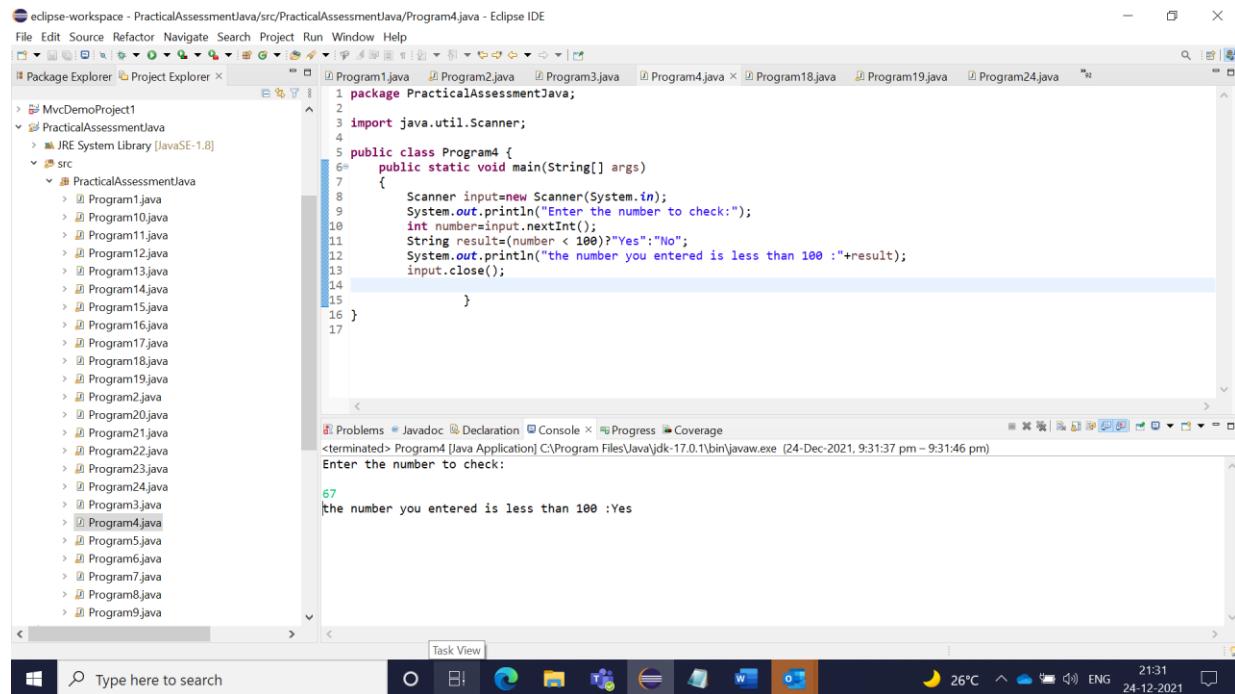


The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "PracticalAssessmentJava" with a "src" folder containing multiple Java files (Program1.java, Program10.java, Program11.java, Program12.java, Program13.java, Program14.java, Program15.java, Program16.java, Program17.java, Program18.java, Program19.java, Program2.java, Program20.java, Program21.java, Program22.java, Program23.java, Program24.java).
- Code Editor:** Displays the content of Program3.java:

```
18 System.out.print(" Please Enter the age of a person, to determine if he/she eligible for covidshots : ");
19
20 if(age > 60)
21 {
22     System.out.println("\n Eligible Now");
23 }
24 else if(age > 45)
25 {
26     System.out.println("\n Eligible in 15 Days");
27 }
28 else if(age > 18)
29 {
30     System.out.println("\n Eligible In a Month");
31 }
32 else
33 {
34     System.out.println("\n Not Eligible");
35 }
36
37 }
```
- Console:** Shows the output of the program: "Please Enter the age of a person, to determine if he/she eligible for covidshots : 15" and "Not Eligible".
- System Tray:** Shows the Windows taskbar with the date and time (24-12-2021, 21:30).

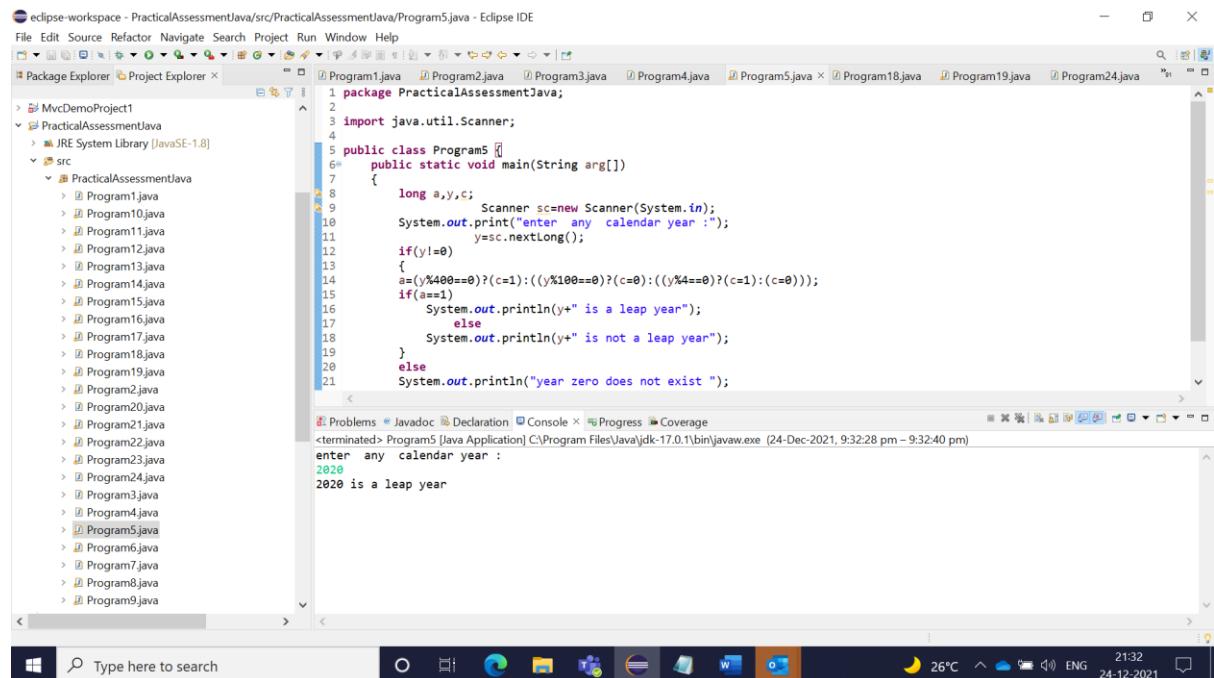
4) Accept input from user and validate if it is a number less than 100 using ternary operator.



```
1 package PracticalAssessmentJava;
2
3 import java.util.Scanner;
4
5 public class Program4 {
6     public static void main(String[] args) {
7         Scanner input=new Scanner(System.in);
8         System.out.println("Enter the number to check:");
9         int number=input.nextInt();
10        String result=(number < 100)?"Yes":"No";
11        System.out.println("the number you entered is less than 100 :"+result);
12        input.close();
13    }
14
15 }
16
17
```

The screenshot shows the Eclipse IDE interface with the code editor open. The code uses a ternary operator to check if the user input is less than 100 and prints the result. The terminal window shows the program running and accepting user input.

5) Accept year from user and validate if it is a leap year using ternary operator



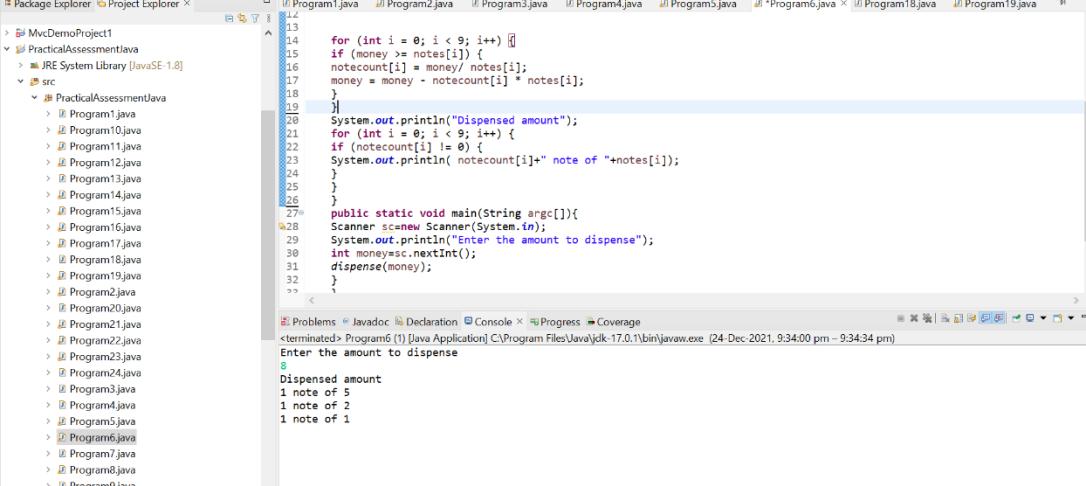
```
1 package PracticalAssessmentJava;
2
3 import java.util.Scanner;
4
5 public class Program5 {
6     public static void main(String arg[]) {
7         long a,y,c;
8         Scanner sc=new Scanner(System.in);
9         System.out.print("enter any calendar year :");
10        y=sc.nextLong();
11        if(y!=0)
12        {
13            a=(y%400==0)?(c=1):((y%100==0)?(c=0):((y%4==0)?(c=1):(c=0)));
14            if(a==1)
15                System.out.println(y+" is a leap year");
16            else
17                System.out.println(y+" is not a leap year");
18        }
19        else
20            System.out.println("year zero does not exist ");
21    }
22
23 }
24
25
```

The screenshot shows the Eclipse IDE interface with the code editor open. The code uses a ternary operator to validate if a user input is a leap year. The terminal window shows the program running and accepting user input.

6) A banker has the following denominations to deliver cash. INR : 1,2,5,10,20,50,100,200,500. Accept a value from user and remit the sum with least number of notes.

**Test cases: In what denominations would a banker remit sum of Rs. 475, Rs 530, Rs 219
Input: integer between 1 and 100,000**

Output: String describing the number of notes and currencies.



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the `MvcDemoProject1` project with a `src` folder containing multiple Java files: `Program1.java`, `Program2.java`, `Program3.java`, `Program4.java`, `Program5.java`, `Program6.java`, `Program18.java`, and `Program19.java`.
- Code Editor:** Displays the content of `Program6.java`. The code is a Java application that takes an amount of money as input and dispenses it using the fewest number of notes possible (5s and 2s). It includes a `main` method and a loop to calculate the number of notes for each denomination.
- Output:** Shows the terminal output of the application, which asks for the amount to dispense and then prints the notes dispensed: "Dispensed amount" followed by "1 note of 5", "1 note of 2", and "1 note of 1".
- Bottom Bar:** Includes the Windows taskbar with pinned icons for File Explorer, File Manager, and Task View, along with the system clock (21:34), battery level (26°C), and network status (ENG 24-12-2021).

7) Accept student total marks M and grade the students based on the marks entered.

0 < M 95 is A+

M > 85 is A

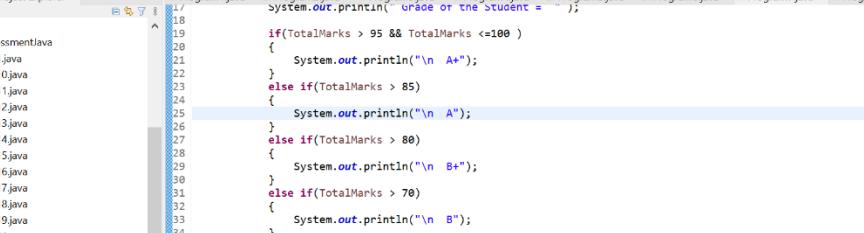
M > 80 is B+

M > 70 is B

M > 60 is C

M > 50 is D

M < 50 Fails.



```
1/    System.out.println("Grade of the Student = ");
2/    if(TotalMarks > 95 & TotalMarks <=100 )
3/    {
4/        System.out.println("\n A+");
5/    }
6/    else if(TotalMarks > 85)
7/    {
8/        System.out.println("\n A");
9/    }
10/   else if(TotalMarks > 80)
11/   {
12/       System.out.println("\n B+");
13/   }
14/   else if(TotalMarks > 70)
15/   {
16/       System.out.println("\n B");
17/   }
18/   else if(TotalMarks > 60)
19/   {
20/       System.out.println("\n B-");
21/   }
22/   else
23/   {
24/       System.out.println("\n C");
25/   }
26/ }
27/ }
28/ }
29/ }
30/ }
31/ }
32/ }
33/ }
34/ }
35/ }
36/ }
37/ }
38/ }
```

Problems Javadoc Declaration Console Coverage

<terminated>Program7 [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (24-Dec-2021, 9:36:59 pm - 9:37:04 pm)

Please Enter the Total Marks :

Grade of the Student =

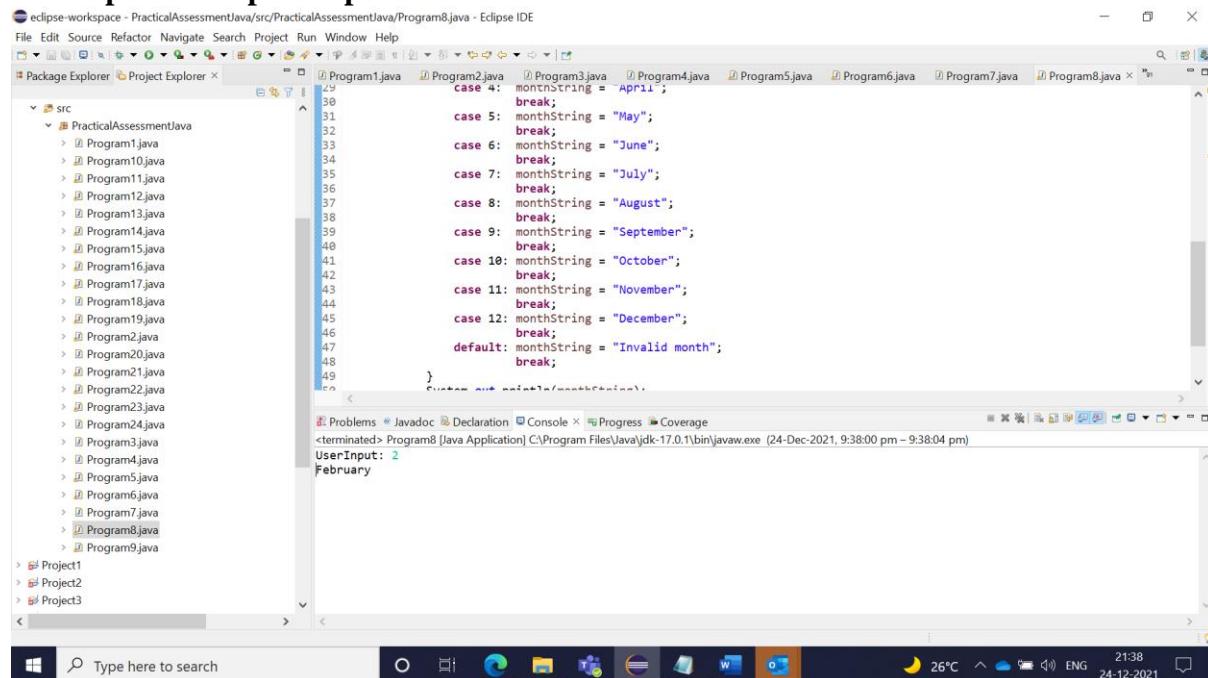
A

8) Using Switch statement: Print the month of the year in literals, based on the numeric value entered.

UserInput: 1

Output: January

UserInput: 9 Output: September



The screenshot shows the Eclipse IDE interface with the following details:

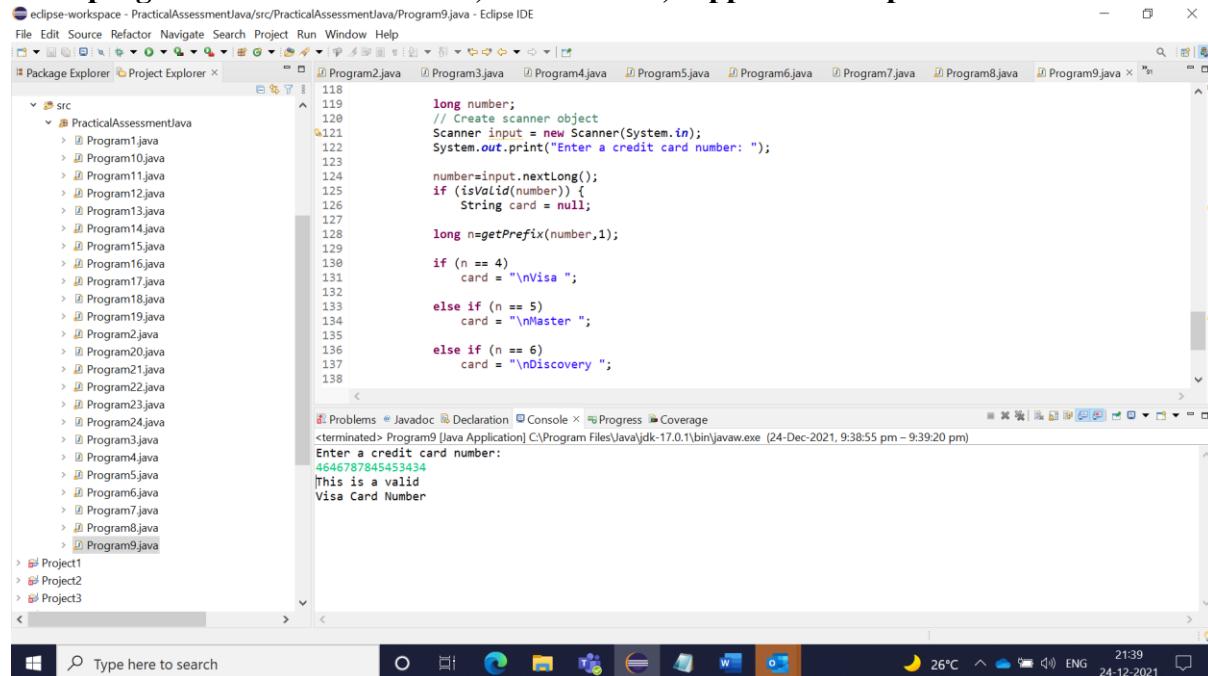
- Project Explorer:** Shows a project named "PracticalAssessmentJava" containing multiple Java files (Program1.java, Program2.java, etc.) and three empty projects (Project1, Project2, Project3).
- Code Editor:** Displays a Java code snippet for a switch statement:

```
case 4: monthString = "April";
break;
case 5: monthString = "May";
break;
case 6: monthString = "June";
break;
case 7: monthString = "July";
break;
case 8: monthString = "August";
break;
case 9: monthString = "September";
break;
case 10: monthString = "October";
break;
case 11: monthString = "November";
break;
case 12: monthString = "December";
break;
default: monthString = "Invalid month";
break;
```
- Console:** Shows the output of the program:

```
UserInput: 2
February
```
- Bottom Bar:** Includes a search bar, taskbar icons, and system status (26°C, ENG, 24-12-2021).

9) Write a function to determine if a given number is a valid credit card number and determine the type of card. All credit card numbers are 16 digits in length. If your credit card numbers begin with 3, then it's always part of the American Express, Diner's Club or Carte Blanche payment networks. If the card begins with a 4, then it is a Visa. Card numbers that begin with 5 are part of the MasterCards, while cards that begin with 6 belong to the Discover network.

This program should have a class, constructors, support user input.



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "PracticalAssessmentJava" containing multiple Java files (Program1.java, Program2.java, etc.) and three empty projects (Project1, Project2, Project3).
- Code Editor:** Displays a Java code snippet for a credit card validation function:

```
long number;
// Create scanner object
Scanner input = new Scanner(System.in);
System.out.print("Enter a credit card number: ");

number=input.nextLong();
if (isValid(number)) {
    String card = null;

    long n=getPrefix(number,1);

    if (n == 4)
        card = "\nVisa ";
    else if (n == 5)
        card = "\nMaster ";
    else if (n == 6)
        card = "\nDiscovery ";
```
- Console:** Shows the output of the program:

```
Enter a credit card number:
4646787845453434
This is a valid
Visa Card Number
```
- Bottom Bar:** Includes a search bar, taskbar icons, and system status (26°C, ENG, 24-12-2021).

10) The following needs String class. Using Switch: 1. Accept a past date from user and check if it is a valid date. Do not use 'SimpleDateFormat' class from java.

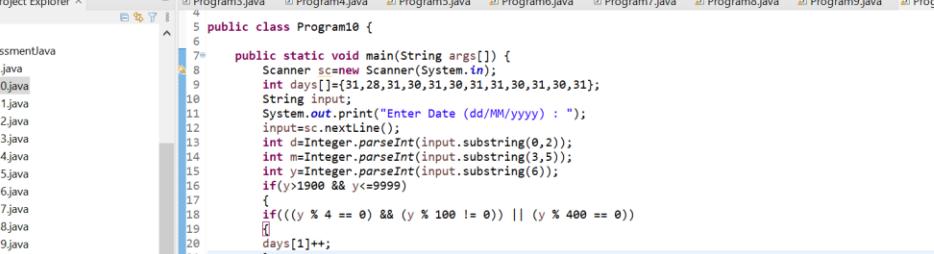
Example Input: Enter

Date(DD/MM/YYYY): 12/12/2009

This is a valid date.

Input: Enter Date(DD/MM/YYYY): 29/02/2009

This is invalid date.



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "PracticalAssessmentJava" containing 23 Java files (Program1.java to Program23.java) and three other projects (Project1, Project2, Project3).
- Package Explorer:** Shows the same structure as the Project Explorer.
- Code Editor:** Displays the content of Program10.java. The code is a Java program that prints the number of days in a given month. It includes a check for leap years and handles invalid input.
- Output:** Shows the terminal output of the program running in the IDE.
- Bottom Bar:** Includes the Windows taskbar with the Eclipse icon, a search bar, and system status icons.

```
4
5 public class Program10 {
6
7     public static void main(String args[]) {
8         Scanner sc=new Scanner(System.in);
9         int days[]={31,28,31,30,31,30,31,31,30,31,30,31};
10        String input;
11        System.out.print("Enter Date (dd/MM/yyyy) : ");
12        inputsc.nextLine();
13        int d=Integer.parseInt(input.substring(0,2));
14        int m=Integer.parseInt(input.substring(3,5));
15        int y=Integer.parseInt(input.substring(6));
16        if(y>1900 && y<=9999)
17        {
18            if((y % 4 == 0) && (y % 100 != 0) || (y % 400 == 0))
19            {
20                days[1]++;
21            }
22            if(m>0 && m<12)
23            {
24                if(d>8 && d< days[m-1])
25                {
26                    if((d-1) % 2 == 0)
27                    {
28                        days[1]++;
29                    }
30                }
31            }
32        }
33        System.out.println("Valid Date");
34    }
35}
```

Problems Javadoc Declaration Console Progress Coverage

<terminated> Program10 [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (24-Dec-2021, 9:40:54 pm - 9:41:07 pm)

Enter Date (dd/MM/yyyy) : 06/04/2021

Valid Date

11) . Can there be return stmt for a constructor?

Ans: No

12) Can we define a non default constructor without define default constructor

Ans: Yes

13) what is wrong with this?

for (int i = 1; i <= 1000; i++)

{

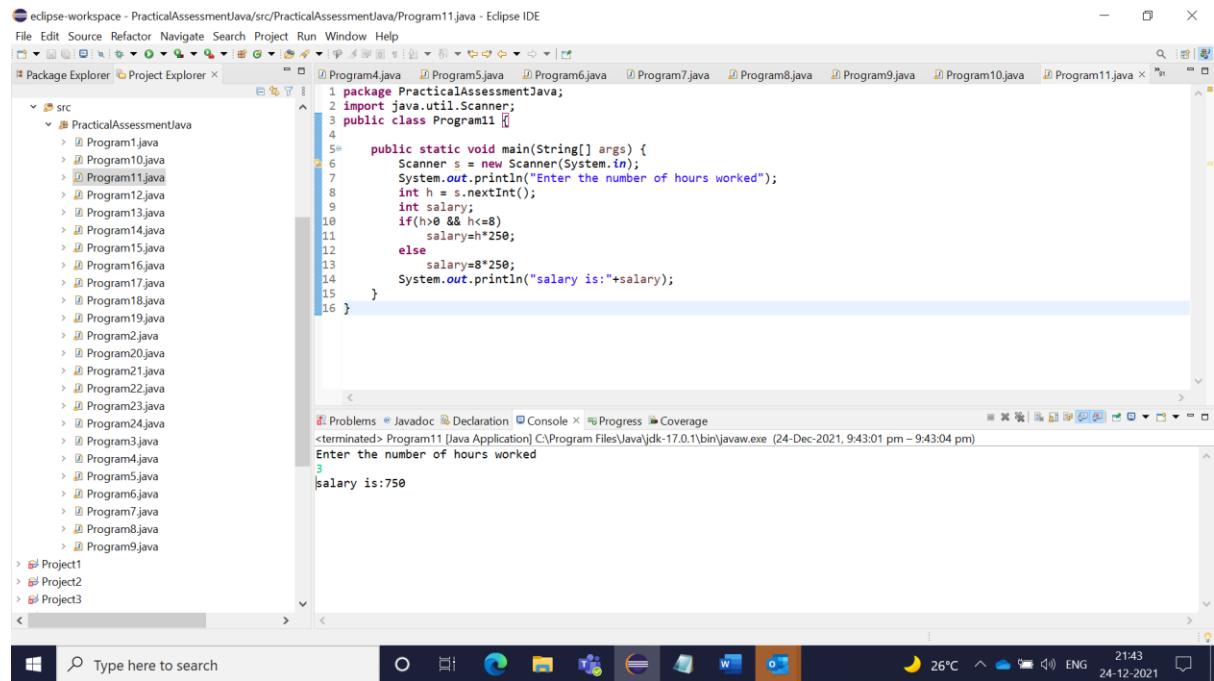
```
int sum = 0; sum = sum + i;
```

}

```
System.out.println("The sum is " + sum);
```

Ans: Sum cannot be resolved to a variable. Because sum is initialized inside for loop and is accessible only inside loop so we cannot print the value of sum.

14) Gather the number of hours employee works, pay only the eight hours he works at the rate of Rs 250 per hour. It is a company policy not to pay over time.



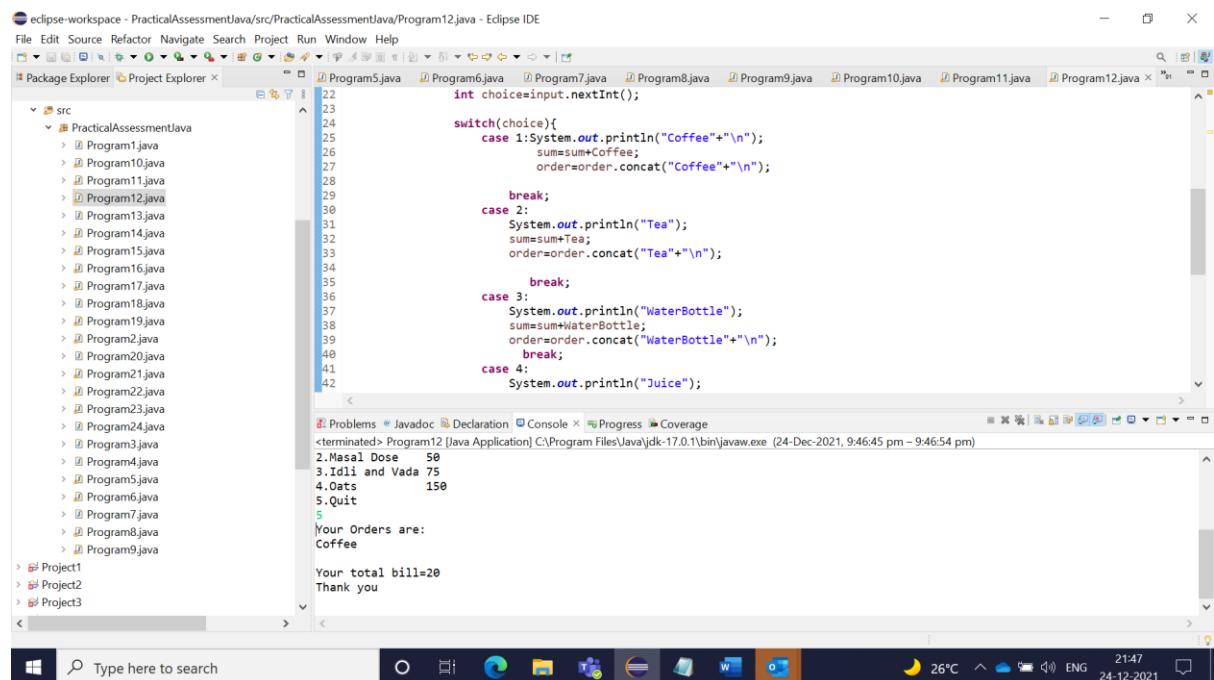
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "PracticalAssessmentJava" with multiple Java files (Program1.java to Program20.java) in the "src" directory.
- Code Editor:** Displays the content of "Program11.java". The code reads a number of hours from the user and calculates a salary of 250 per hour, capped at 8 hours.

```
1 package PracticalAssessmentJava;
2 import java.util.Scanner;
3 public class Program11 {
4
5     public static void main(String[] args) {
6         Scanner s = new Scanner(System.in);
7         System.out.println("Enter the number of hours worked");
8         int h = s.nextInt();
9         int salary;
10        if(h>0 && h<=8)
11            salary=h*250;
12        else
13            salary=8*250;
14        System.out.println("salary is:"+salary);
15    }
16 }
```

- Console:** Shows the output of the program: "Enter the number of hours worked" followed by "salary is:750".
- System Tray:** Shows the date (24-12-2021), time (21:43), and weather (26°C).

15) Present a menu of a restaurant and calculate the cost of breakfast.



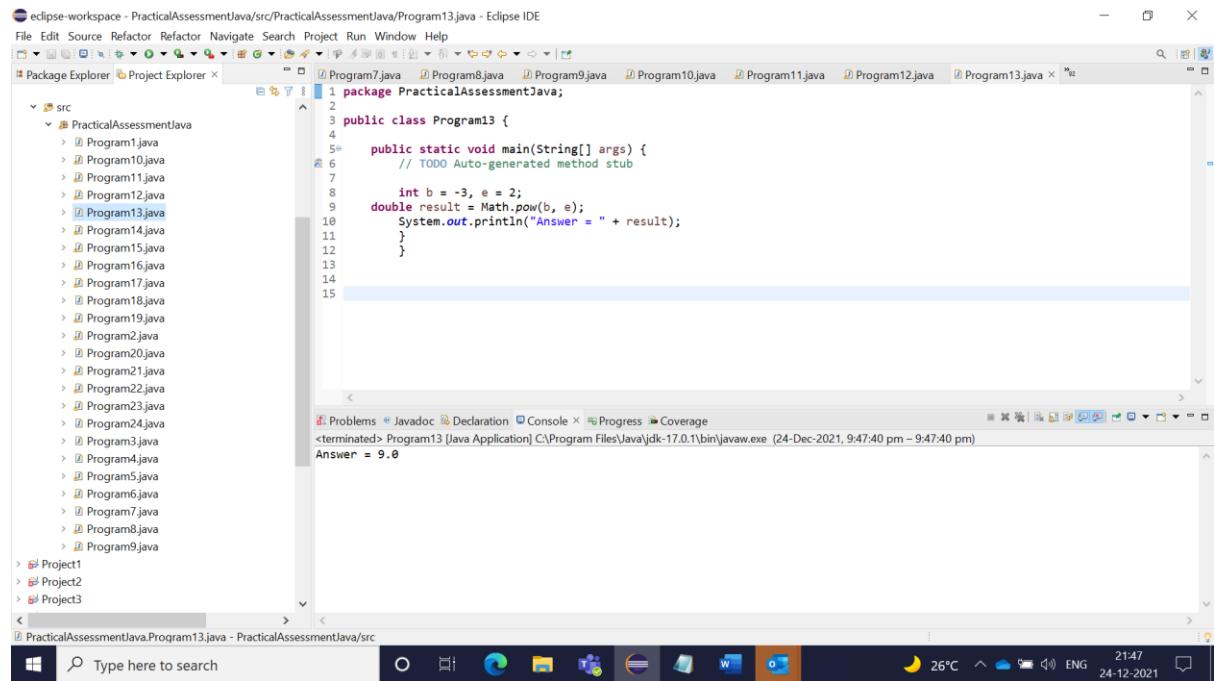
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "PracticalAssessmentJava" with multiple Java files (Program1.java to Program20.java) in the "src" directory.
- Code Editor:** Displays the content of "Program12.java". The code presents a menu for breakfast items (Coffee, Tea, WaterBottle, Juice) and calculates a total bill based on user input.

```
1 package PracticalAssessmentJava;
2 import java.util.Scanner;
3 public class Program12 {
4
5     public static void main(String[] args) {
6         int choice=input.nextInt();
7
8         switch(choice){
9             case 1:System.out.println("Coffee"+'\n');
10            sumsum+=Coffee;
11            order=order.concat("Coffee"+'\n');
12
13            break;
14            case 2:
15                System.out.println("Tea");
16                sumsum+=Tea;
17                order=order.concat("Tea"+'\n');
18
19                break;
20            case 3:
21                System.out.println("WaterBottle");
22                sumsum+=WaterBottle;
23                order=order.concat("WaterBottle"+'\n');
24
25                break;
26            case 4:
27                System.out.println("Juice");
28
29        }
30
31        System.out.println("Your Orders are:");
32        System.out.println("Coffee");
33
34        System.out.println("Your total bill=");
35        System.out.println(sumsum);
36
37        System.out.println("Thank you");
38
39    }
40
41 }
```

- Console:** Shows the output of the program: "Your Orders are: Coffee" followed by the total bill (e.g., "Your total bill=20").
- System Tray:** Shows the date (24-12-2021), time (21:47), and weather (26°C).

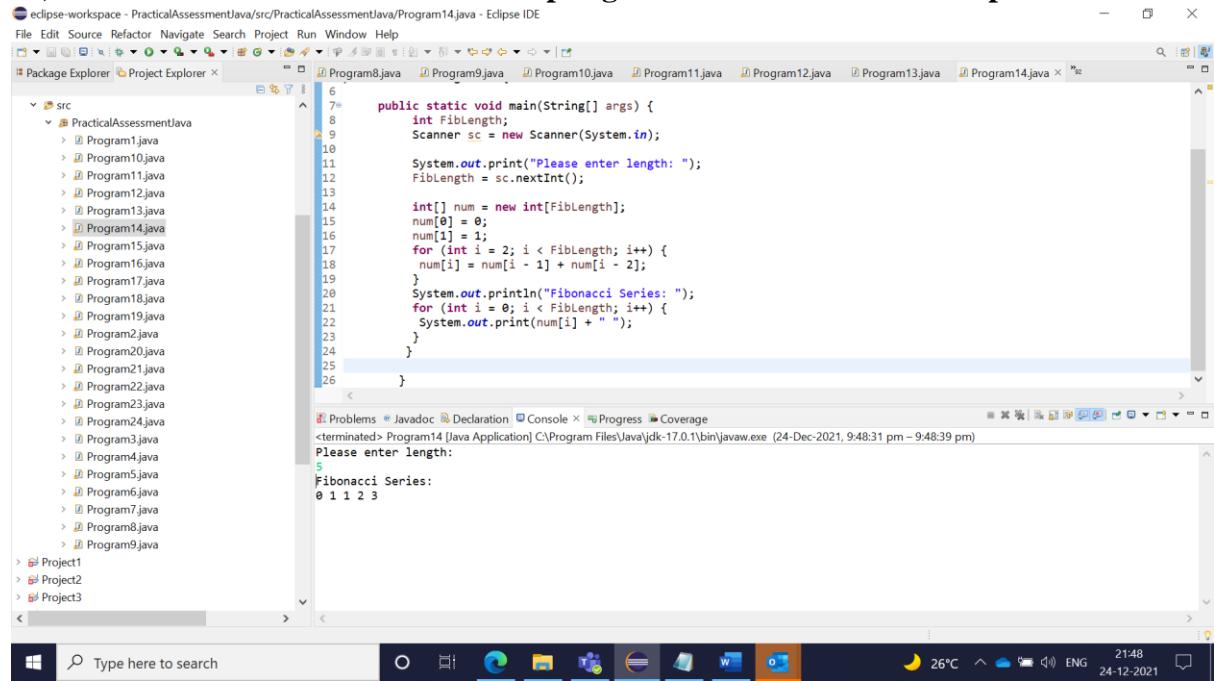
16) Accept a base b, and an exponent e. write a function power(b,e) that returns b raised e. Print the solution.



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "PracticalAssessmentJava" containing 21 Java files (Program1.java to Program21.java) and three empty projects (Project1, Project2, Project3).
- Code Editor:** Displays the content of `Program13.java` which contains a `main` method that calculates b^e using `Math.pow`.
- Console:** Shows the output of the program: `Answer = 9.0`.
- System Tray:** Shows the date (24-12-2021), time (21:47), and system status (26°C, ENG).

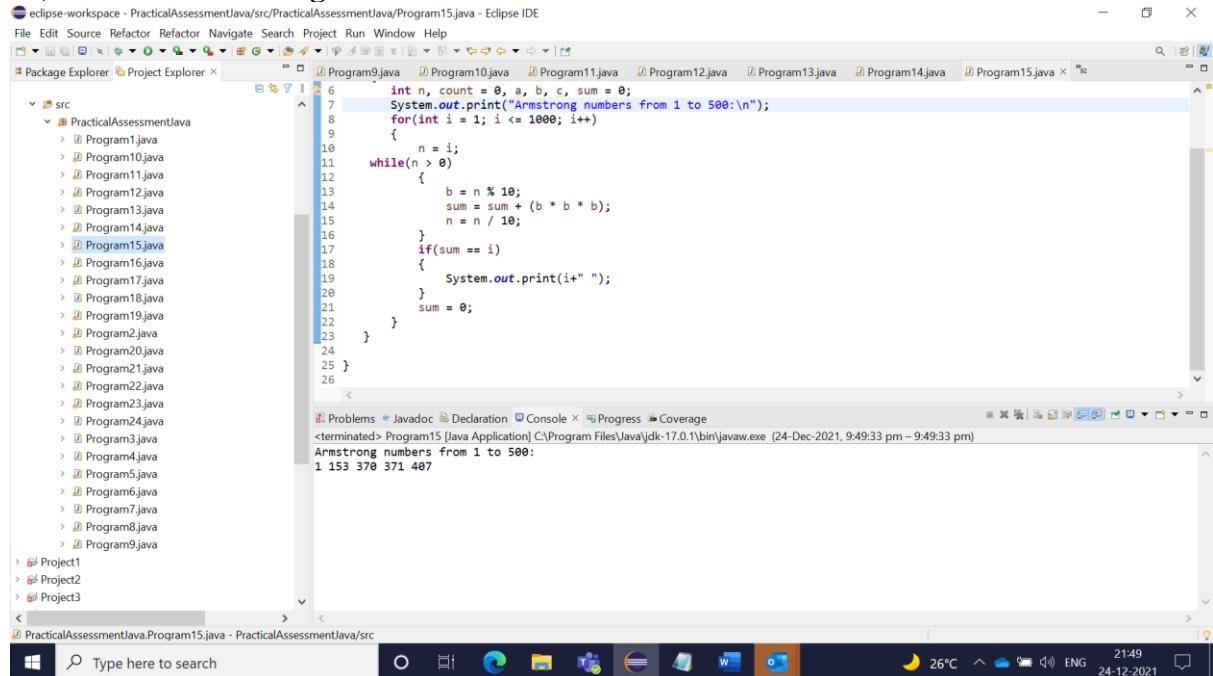
17) Generate fibonacci terms after accepting the number of terms to be printed.



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "PracticalAssessmentJava" containing 21 Java files (Program1.java to Program21.java) and three empty projects (Project1, Project2, Project3).
- Code Editor:** Displays the content of `Program14.java` which prompts the user for the length of the Fibonacci series and prints the series.
- Console:** Shows the output of the program: `Please enter length: 5` and the resulting Fibonacci Series: `0 1 1 2 3`.
- System Tray:** Shows the date (24-12-2021), time (21:48), and system status (26°C, ENG).

18) Find all Armstrong numbers from 1 to 500.



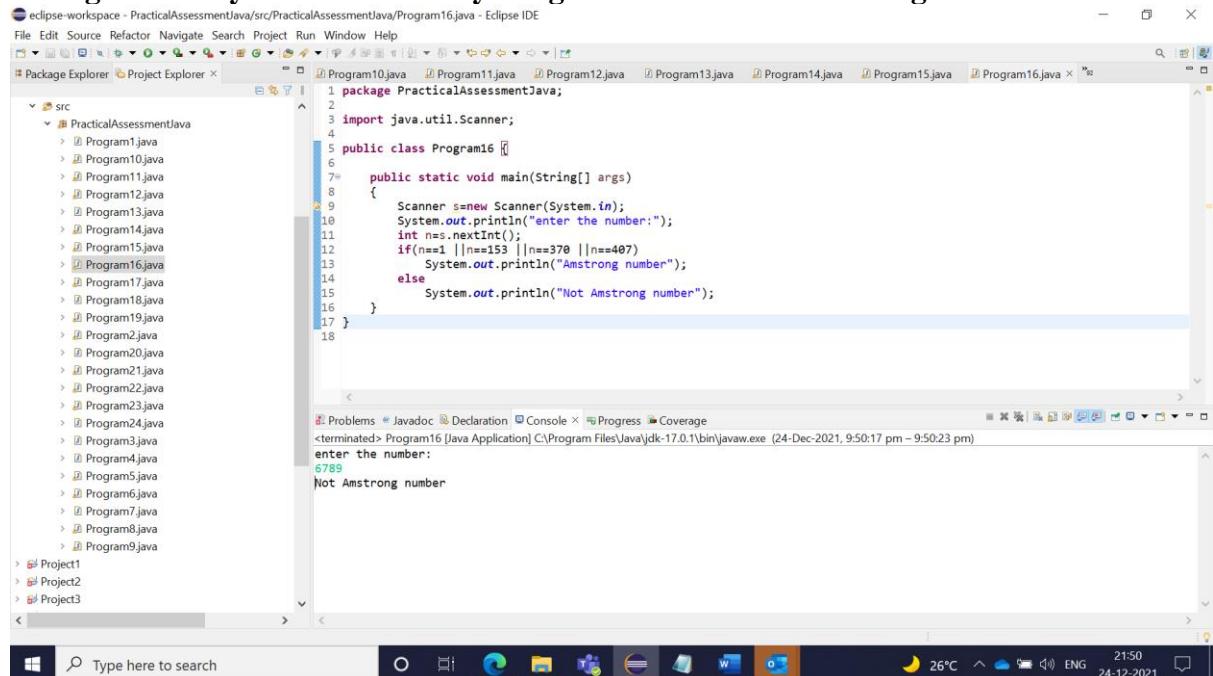
The screenshot shows the Eclipse IDE interface with the following details:

- File Menu:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Project Explorer:** Shows a project named "PracticalAssessmentJava" with a "src" folder containing 20 Java files (Program1.java to Program20.java).
- Code Editor:** Displays the content of Program15.java. The code prints Armstrong numbers from 1 to 500. The output in the Console shows:

```
Armstrong numbers from 1 to 500:  
1 153 370 371 407
```
- Console:** Shows the output of the program: "Armstrong numbers from 1 to 500: 1 153 370 371 407".
- Bottom Bar:** Shows the Windows taskbar with the Eclipse icon, a search bar, and system status (26°C, ENG, 24-12-2021).

19) An Armstrong number is a n-digit number that is equal to the sum of nth power of its digits. Example all single digit number are Armstrong numbers 3 to power 1 is 3 371 = 3 Power 3 + 7 power 3 + 1 power 3 = 371.

Changes: Modify the code to verify if a given number is Armstrong number



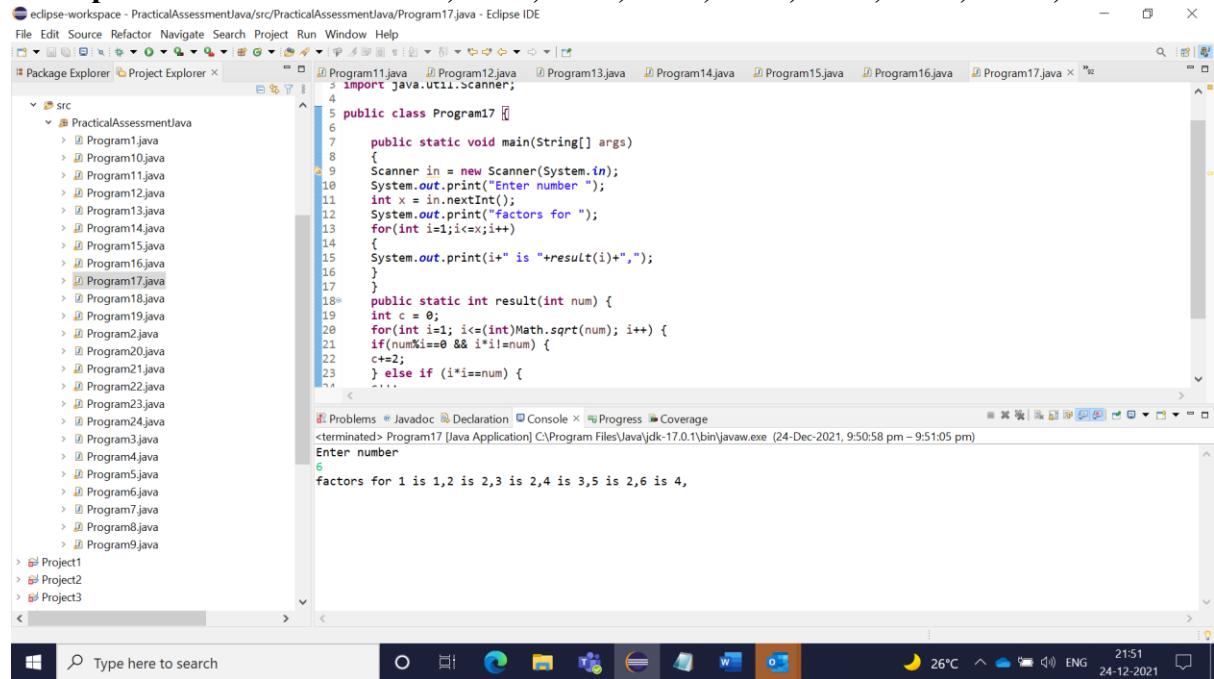
The screenshot shows the Eclipse IDE interface with the following details:

- File Menu:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Project Explorer:** Shows a project named "PracticalAssessmentJava" with a "src" folder containing 20 Java files (Program1.java to Program20.java).
- Code Editor:** Displays the content of Program16.java. The code uses a Scanner to read a number and checks if it is an Armstrong number. The output in the Console shows:

```
enter the number:  
6789  
Not Amstrong number
```
- Console:** Shows the output of the program: "enter the number: 6789 Not Amstrong number".
- Bottom Bar:** Shows the Windows taskbar with the Eclipse icon, a search bar, and system status (26°C, ENG, 24-12-2021).

20) Count the number of factors for every number from 1 to N.

Example: N = 9 Facots for 1 is 1, 2 is 2, 3 is 2, 4 is 3, 5 is 2, 6 is 4, 7 is 2, 8 is 4 , 9 is 3



The screenshot shows the Eclipse IDE interface with the following details:

- File Menu:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard Eclipse toolbar with icons for file operations.
- Package Explorer:** Shows a project named "PracticalAssessmentJava" with multiple Java files (Program1.java to Program18.java) listed under the "src" folder.
- Project Explorer:** Shows three projects: Project1, Project2, and Project3.
- Code Editor:** The "Program17.java" file is open, containing the following Java code:

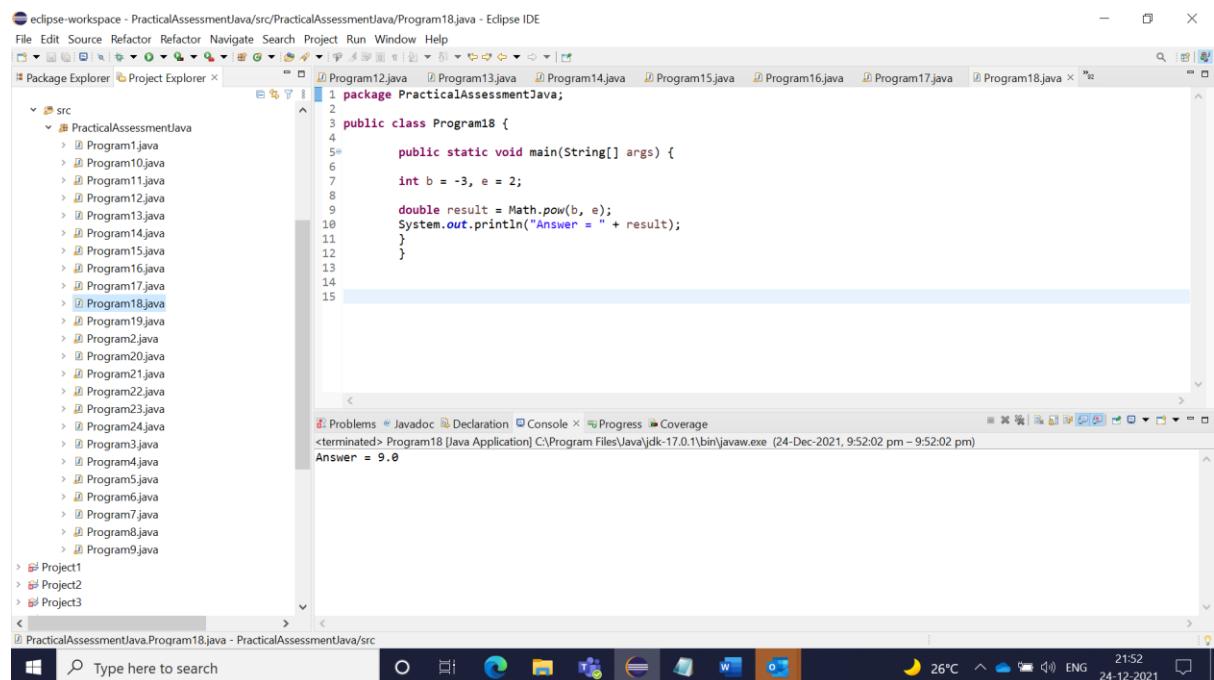
```
1 package PracticalAssessmentJava;
2
3 public class Program17 {
4
5     public static void main(String[] args) {
6
7         Scanner in = new Scanner(System.in);
8
9         Scanner in = new Scanner(System.in);
10        System.out.print("Enter number ");
11        int x = in.nextInt();
12        System.out.print("Factors for ");
13        for(int i=1;i<=x;i++) {
14
15            System.out.print(i+" is "+result(i)+",");
16        }
17    }
18
19    public static int result(int num) {
20        int c = 0;
21        for(int i=1; i<=(int) Math.sqrt(num); i++) {
22            if(num%i==0 && i!=1&num) {
23                c+=2;
24            } else if (i*i==num) {
25                c+=1;
26            }
27        }
28        return c;
29    }
30
31 }
```

- Output View:** Shows the terminal output of the program execution:

```
Enter number
6
Factors for 1 is 1,2 is 2,3 is 2,4 is 3,5 is 2,6 is 4,
```

- System Tray:** Shows the Windows taskbar with the date (24-12-2021), time (21:51), and system status (26°C, ENG).

21) Accept a Base b, and an exponent e. write a function power(b,e) that returns b raised e. Print the solution.



The screenshot shows the Eclipse IDE interface with the following details:

- File Menu:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard Eclipse toolbar with icons for file operations.
- Package Explorer:** Shows a project named "PracticalAssessmentJava" with multiple Java files (Program1.java to Program18.java) listed under the "src" folder.
- Project Explorer:** Shows three projects: Project1, Project2, and Project3.
- Code Editor:** The "Program18.java" file is open, containing the following Java code:

```
1 package PracticalAssessmentJava;
2
3 public class Program18 {
4
5     public static void main(String[] args) {
6
7         int b = -3, e = 2;
8
9         double result = Math.pow(b, e);
10        System.out.println("Answer = " + result);
11    }
12
13
14
15 }
```

- Output View:** Shows the terminal output of the program execution:

```
Answer = 9.0
```

- System Tray:** Shows the Windows taskbar with the date (24-12-2021), time (21:52), and system status (26°C, ENG).

22) HCF: Highest Common Factor.

The HCF of two numbers is the greatest number that divides the two numbers.

Example: HCF of (14, 42) is 14.

Factors of 14 -> 2, 7, 14 Factors of 42 -> 2, 3, 6, 7, 14, 21

23) . Print the following pattern on the screen.

* * * * *



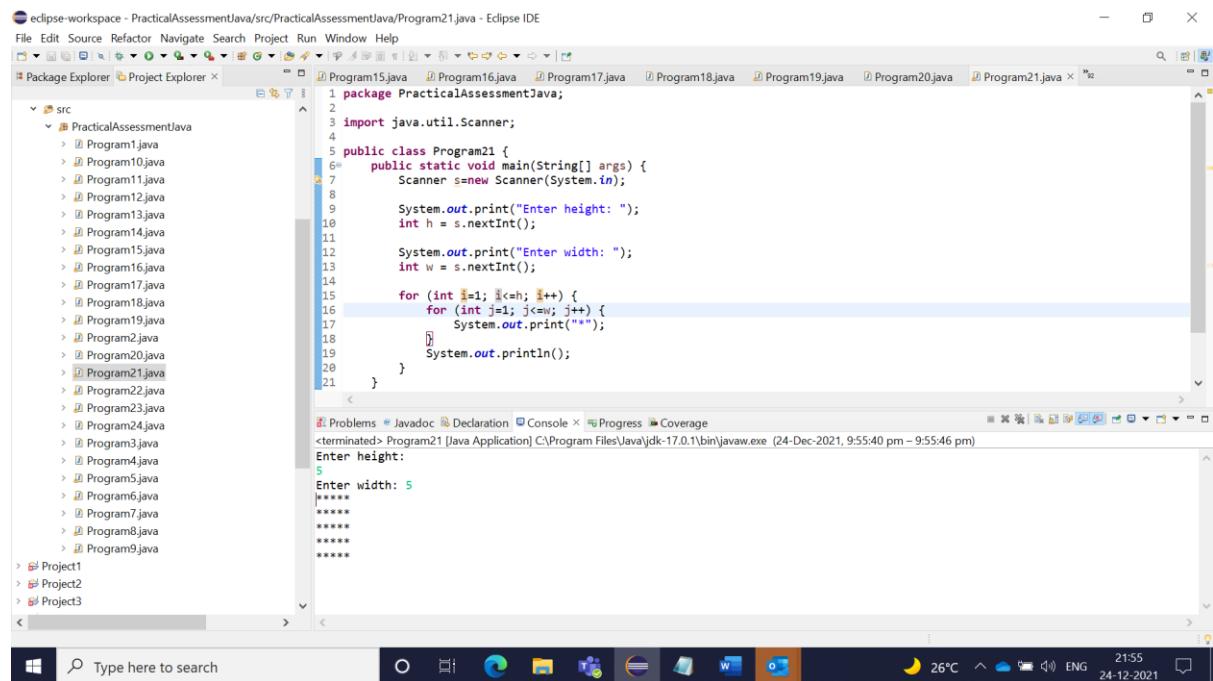
The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the 'PracticalAssessmentJava' project with files: Program1.java, Program10.java, Program11.java, Program12.java, Program13.java, Program14.java, Program15.java, Program16.java, Program17.java, Program18.java, Program19.java, and Program20.java.
- Code Editor:** Displays the content of Program20.java. The code prints a 5x5 grid of asterisks using nested for loops.
- Output View:** Shows the terminal output of the program, which is a 5x5 grid of asterisks:

```
*****
 * * *
 * * * *
 * * * * *
```
- Bottom Bar:** Includes the Windows taskbar with the Eclipse icon, a search bar, and system icons for date, time, battery, and language.

24) Accept height and width and print the pattern with a '*' Ex for height = 5 , width = 5,

```
*****
*****
*****
*****
*****
```

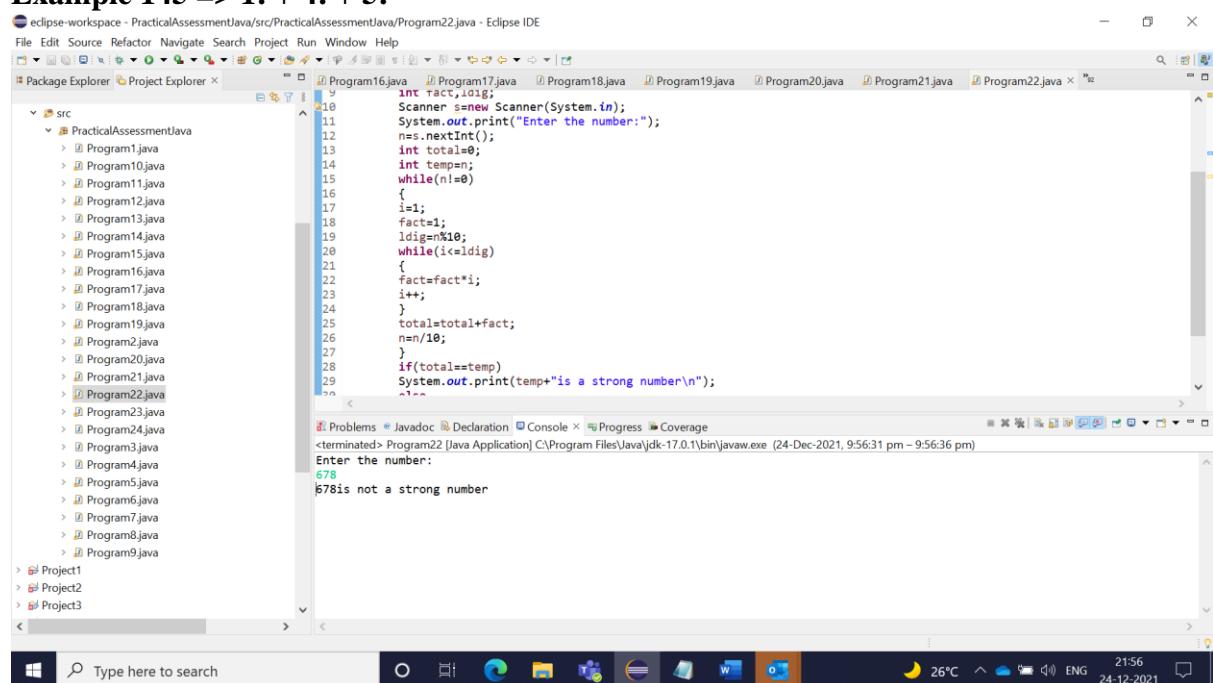


The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "PracticalAssessmentJava" with multiple Java files (Program1.java to Program21.java) in the "src" directory.
- Code Editor:** Displays the content of Program21.java, which contains code to read height and width from the user and print a diamond pattern of asterisks. The code is as follows:1 package PracticalAssessmentJava;
2
3 import java.util.Scanner;
4
5 public class Program21 {
6 public static void main(String[] args) {
7 Scanner s=new Scanner(System.in);
8
9 System.out.print("Enter height: ");
10 int h = s.nextInt();
11
12 System.out.print("Enter width: ");
13 int w = s.nextInt();
14
15 for (int i=1; i<=h; i++) {
16 for (int j=1; j<=i; j++) {
17 System.out.print("*");
18 }
19 System.out.println();
20 }
21 }
22 }
- Console:** Shows the output of the program. The user enters "5" for height and "5" for width. The output is a diamond pattern of 5 rows of asterisks.
- System Tray:** Shows the Windows taskbar with the date and time as 24-12-2021 21:55.

25) Definition of strong number: The sum of the factorials of digits of a number should add upto the number itself.

Example 145 => $1! + 4! + 5!$



The screenshot shows the Eclipse IDE interface with the following details:

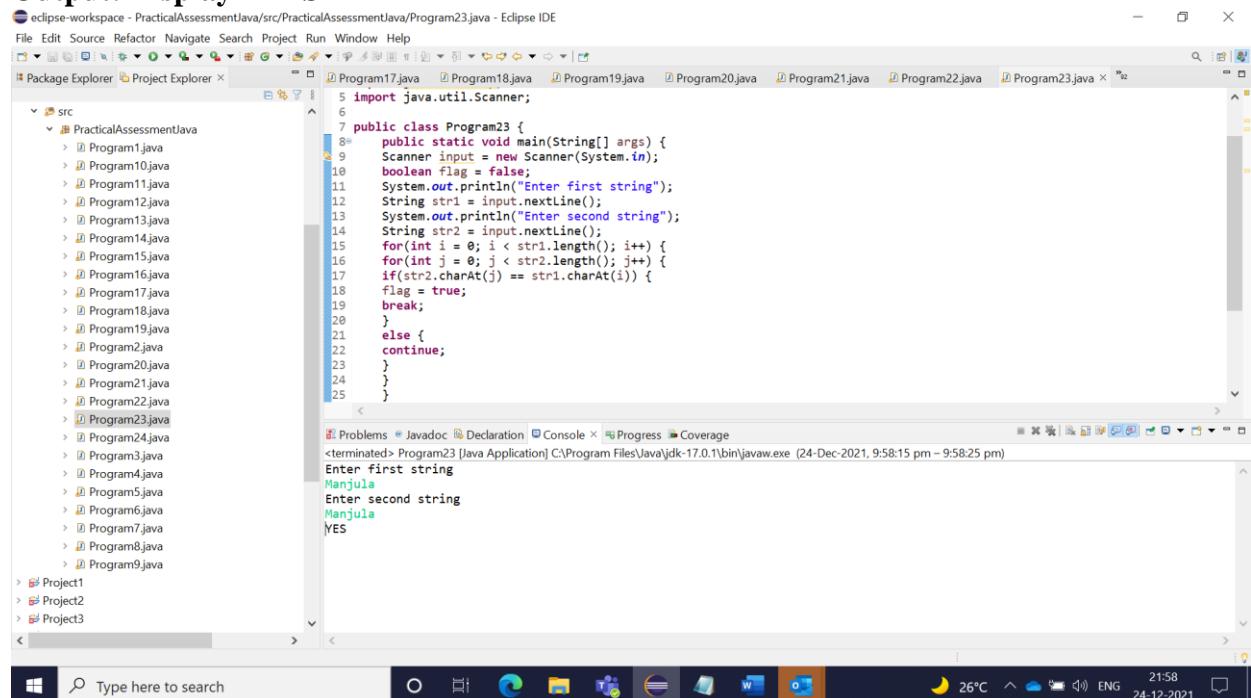
- Project Explorer:** Shows a project named "PracticalAssessmentJava" with multiple Java files (Program1.java to Program21.java) in the "src" directory.
- Code Editor:** Displays the content of Program22.java, which contains code to read a number from the user and check if it is a strong number. The code is as follows:1 package PracticalAssessmentJava;
2
3 import java.util.Scanner;
4
5 public class Program22 {
6 public static void main(String[] args) {
7 int fact,ldig;
8 Scanner s=new Scanner(System.in);
9 System.out.print("Enter the number:");
10 int total=0;
11 int tempn;
12 while(n!=0)
13 {
14 i=1;
15 fact=1;
16 ldig=n%10;
17 while(i<=ldig)
18 {
19 fact=fact*i;
20 i++;
21 }
22 total=total+fact;
23 n=n/10;
24 }
25 if(total==temp)
26 System.out.print(temp+" is a strong number\n");
27 }
28 }
- Console:** Shows the output of the program. The user enters "678" and the program prints "678 is not a strong number".
- System Tray:** Shows the Windows taskbar with the date and time as 24-12-2021 21:56.

26) Statement: Find if there is any character common between the two strings.

If there is a common character return YES else return "NO"

UserInputs: 1. Accept two strings from user.

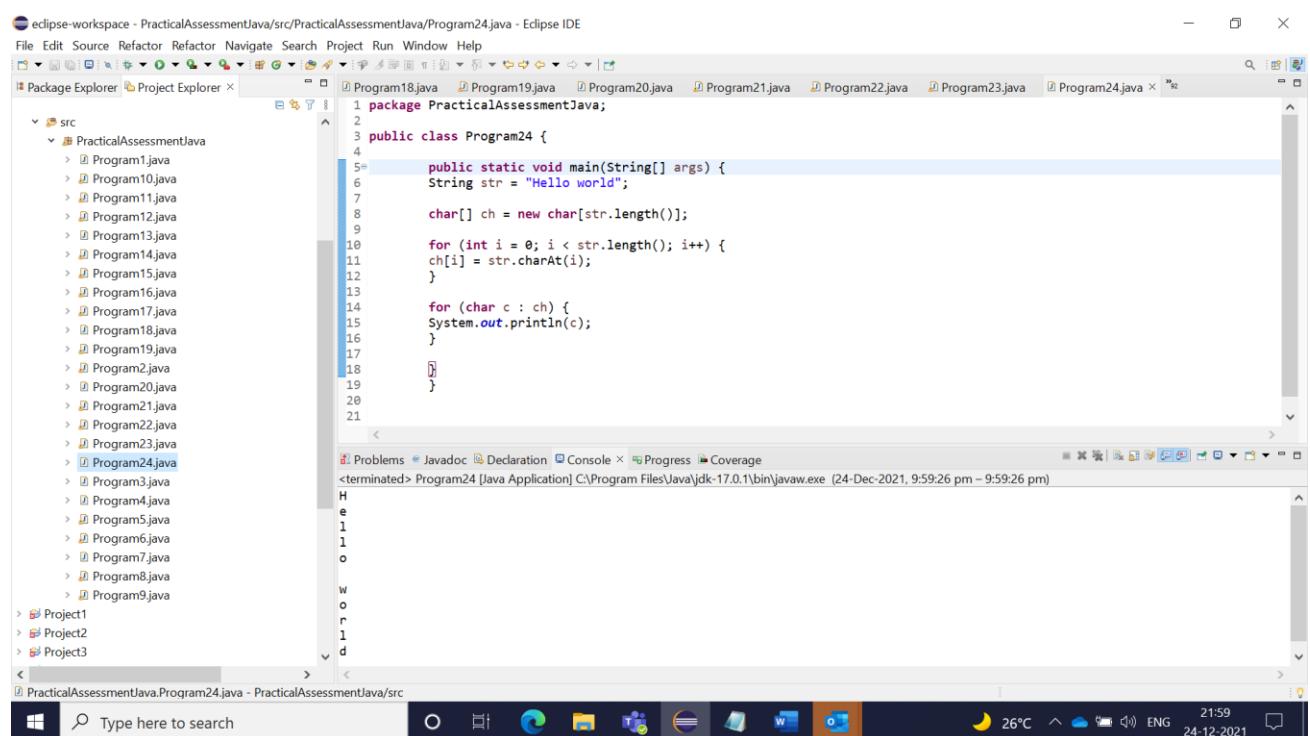
Output: Display "YES"



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "PracticalAssessmentJava" with multiple Java files (Program1.java to Program23.java) under the "src" folder.
- Code Editor:** Displays the content of "Program23.java". The code uses Scanner to read two strings from the user and then compares their characters to determine if there is any common character.
- Console:** Shows the output of the program. The user enters "Manjula" as the first string and "Manjula" as the second string. The output "YES" is displayed, indicating that there is a common character ('m') between the two strings.
- System Tray:** Shows the Windows taskbar with various pinned icons and system status.

27) Convert string to char array



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows a project named "PracticalAssessmentJava" with multiple Java files (Program1.java to Program24.java) under the "src" folder.
- Code Editor:** Displays the content of "Program24.java". The code creates a character array from the string "Hello world" and prints each character to the console.
- Console:** Shows the output of the program. The string "Hello world" is printed character by character: H, e, l, l, o, w, o, r, l, d.
- System Tray:** Shows the Windows taskbar with various pinned icons and system status.