

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- Left Sidebar:** Package Explorer, Console (terminated output), Progress.
- Right Sidebar:** Standard Eclipse sidebar icons.
- Central Area:** Two tabs: Application.java and Practical\_Assessment.java.
- Code Editor (Practical\_Assessment.java):**

```
1 package assessment;
2
3 public class Practical_Assessment {
4     public static void main(String a[]) {
5         System.out.println("1. Hello World.\b Hello World.");
6         System.out.println("2. Hello World.\t Hello World.");
7         System.out.println("3. Hello World.\n Hello World.");
8         System.out.println("4. Hello World.\n Hello World.");
9         System.out.println("5. Hello World.\r Hello World.");
10        System.out.println("6. Hello World.\\" Hello World.");
11        System.out.println("7. Hello World.\' Hello World.");
12        System.out.println("8. Hello World.\\ Hello World.");
13    }
14 }
```
- Console Tab Output:**

```
<terminated> Practical_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.justj.openj
1. Hello World.\b Hello World.
2. Hello World. Hello World.
3. Hello World.
Hello World.
4. Hello World.
Hello World.
5. Hello World.
Hello World.
6. Hello World." Hello World.
7. Hello World.' Hello World.
8. Hello World.\ Hello World.
```
- Bottom Status Bar:** Writable, Smart Insert, 5:60 : 153.



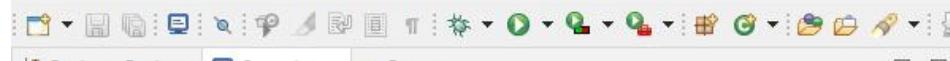
Type here to search



5:21 PM  
26°C  
12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open

Enter a number:

76

76 is an even number

```
1 package assessment;
2
3 //Accept a number from user and determine if a given number is odd or even and print.
4
5 import java.util.Scanner;
6
7 public class Practical_Assessment {
8     public static void main(String a[]) {
9         Scanner sc=new Scanner(System.in);
10        System.out.println("Enter a number:");
11        int num=sc.nextInt();
12        if (num%2==0) {
13            System.out.print(num+" is an even number\n");
14        }
15        else {
16            System.out.print(num+" is an odd number\n");
17        }
18    }
19}
```



Type here to search



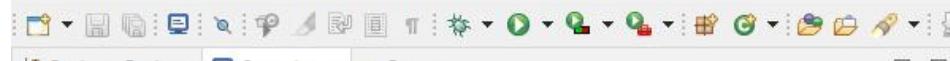
26°C

5:27 PM 12/24/2021



# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



```
<terminated> Practical_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open
```

Enter a number:

45

45 is an odd number

```
1 package assessment;
2
3 //Accept a number from user and determine if a given number is odd or even and print.
4
5 import java.util.Scanner;
6
7 public class Practical_Assessment {
8     public static void main(String a[]) {
9         Scanner sc=new Scanner(System.in);
10        System.out.println("Enter a number:");
11        int num=sc.nextInt();
12        if (num%2==0) {
13            System.out.print(num+" is an even number\n");
14        }
15        else {
16            System.out.print(num+" is an odd number\n");
17        }
18    }
19 }
20
```

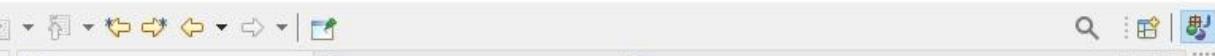


Type here to search



# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Package Explorer Console X Progress

<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open

Enter your age:

45

Eligible in a month

```
1 package assessment;
2
3 //Accept age of a person and determine if he/she eligible for covidshots.
4 //1. More than 60, print "Eligible now"
5 //2. More than 45, print "Eligible in 15 days"
6 //3. More than 18, print "Eligible in a month"
7 //4. Less than 18, print "Not Eligible"
8
9
10 import java.util.Scanner;
11
12 public class Practical_Assessment {
13     public static void main(String a[]) {
14         Scanner sc=new Scanner(System.in);
15         System.out.println("Enter your age:");
16         int age=sc.nextInt();
17         if (age>60) {
18             System.out.println(" Eligible now");
19         }
20         else if (age<=60 && age>45){
21             System.out.println("Eligible in 15 days");
22         }
23         else if (age<=45 && age>18){
24             System.out.println("Eligible in a month");
25         }
26         else if (age<18){
27             System.out.println("Not Eligible");
28         }
29     }
30 }
31 }
```



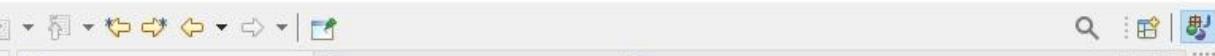
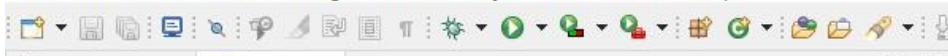
Type here to search



5:36 PM  
25°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Package Explorer Console X Progress

<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open

Enter your age:

10

Not Eligible

```
1 package assessment;
2
3 //Accept age of a person and determine if he/she eligible for covidshots.
4 //1. More than 60, print "Eligible now"
5 //2. More than 45, print "Eligible in 15 days"
6 //3. More than 18, print "Eligible in a month"
7 //4. Less than 18, print "Not Eligible"
8
9
10 import java.util.Scanner;
11
12 public class Practical_Assessment {
13     public static void main(String a[]) {
14         Scanner sc=new Scanner(System.in);
15         System.out.println("Enter your age:");
16         int age=sc.nextInt();
17         if (age>60) {
18             System.out.println(" Eligible now");
19         }
20         else if (age<=60 && age>45){
21             System.out.println("Eligible in 15 days");
22         }
23         else if (age<=45 && age>18){
24             System.out.println("Eligible in a month");
25         }
26         else if (age<18){
27             System.out.println("Not Eligible");
28         }
29     }
30 }
31 }
```



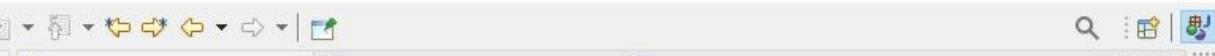
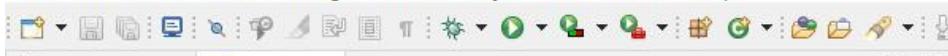
Type here to search



5:36 PM  
25°C  
12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Package Explorer Console X Progress

<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open

Enter your age:

61

| Eligible now

```
1 package assessment;
2
3 //Accept age of a person and determine if he/she eligible for covidshots.
4 //1. More than 60, print "Eligible now"
5 //2. More than 45, print "Eligible in 15 days"
6 //3. More than 18, print "Eligible in a month"
7 //4. Less than 18, print "Not Eligible"
8
9
10 import java.util.Scanner;
11
12 public class Practical_Assessment {
13     public static void main(String a[]) {
14         Scanner sc=new Scanner(System.in);
15         System.out.println("Enter your age:");
16         int age=sc.nextInt();
17         if (age>60) {
18             System.out.println(" Eligible now");
19         }
20         else if (age<=60 && age>45){
21             System.out.println("Eligible in 15 days");
22         }
23         else if (age<=45 && age>18){
24             System.out.println("Eligible in a month");
25         }
26         else if (age<18){
27             System.out.println("Not Eligible");
28         }
29     }
30 }
31 }
```



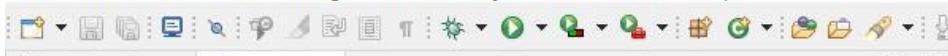
Type here to search



5:36 PM  
25°C  
12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Package Explorer Console X Progress

<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open

Enter your age:

53

Eligible in 15 days

```
1 package assessment;
2
3 //Accept age of a person and determine if he/she eligible for covidshots.
4 //1. More than 60, print "Eligible now"
5 //2. More than 45, print "Eligible in 15 days"
6 //3. More than 18, print "Eligible in a month"
7 //4. Less than 18, print "Not Eligible"
8
9
10 import java.util.Scanner;
11
12 public class Practical_Assessment {
13     public static void main(String a[]) {
14         Scanner sc=new Scanner(System.in);
15         System.out.println("Enter your age:");
16         int age=sc.nextInt();
17         if (age>60) {
18             System.out.println(" Eligible now");
19         }
20         else if (age<=60 && age>45){
21             System.out.println("Eligible in 15 days");
22         }
23         else if (age<=45 && age>18){
24             System.out.println("Eligible in a month");
25         }
26         else if (age<18){
27             System.out.println("Not Eligible");
28         }
29     }
30 }
31 }
```



Type here to search



5:37 PM

25°C

12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project navigation.
- Left Sidebar:** Package Explorer, Console (containing output), and Progress views.
- Top Status Bar:** Shows the current file path: Practical\_Assessment/src/assessment/Practical\_Assessment.java.
- Central Area:** A code editor window displaying Java code. The code defines a class named Practical\_Assessment with a main method that reads a user input, validates it, and prints the result.

```
1 package assessment;
2
3 //Accept input from user and validate if it is a number less than 100 using ternary operator.
4
5
6 import java.util.Scanner;
7
8 public class Practical_Assessment {
9     public static void main(String[] args) {
10         Scanner num = new Scanner(System.in);
11         System.out.println("Enter your Number : ");
12         int number = num.nextInt();
13         String result = (number<100) ? "Validate":"invalidate";
14         System.out.println("Your Number is :" +result);
15         num.close();
16     }
17 }
18
```

- Bottom Status Bar:** Shows system information including the date (12/24/2021), time (7:24 PM), battery level (12%), and network status.



eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project navigation.
- Views:** Package Explorer, Console, and Progress are visible in the top left.
- Project Explorer:** Shows a project named "Practical\_Assessment" containing files like Date.java, \*Amstrong.java, Method3.java, and Employee\_hou...
- Console View:** Displays the output of the application run. It shows the application prompting for input ("Enter your Number :"), receiving the input "101", and then outputting "Your Number is :invalidate".
- Code Editor:** The main editor window displays the Java code for the "Practical\_Assessment" class. The code uses a Scanner to read an integer from standard input, prints a prompt to standard output, reads the integer, and then prints the result based on whether it's less than 100 or not. The code is annotated with line numbers (1 through 18) and syntax highlighting.

```
1 package assessment;
2
3 //Accept input from user and validate if it is a number less than 100 using ternary operator.
4
5
6 import java.util.Scanner;
7
8 public class Practical_Assessment {
9     public static void main(String[] args) {
10         Scanner num = new Scanner(System.in);
11         System.out.println("Enter your Number : ");
12         int number = num.nextInt();
13         String result = (number<100) ? "Validate":"invalidate";
14         System.out.println("Your Number is :" +result);
15         num.close();
16     }
17 }
18
```



Type here to search



7:24 PM  
23°C  
12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- Left Sidebar:** Package Explorer, Console (terminated output), Progress.
- Right Sidebar:** Shows other open Java files: Scanner\_class.java, Date.java, \*Amstrong.java, Employee\_hours.java.
- Code Editor:** Displays Java code for a leap year checker. The code uses a scanner to input a year and prints whether it is a leap year or not based on specific rules (divisibility by 400, 100, or 4). The code editor has syntax highlighting and line numbers.
- Status Bar:** Writable, Smart Insert, 3:81:103.



Type here to search



7:26 PM  
23°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



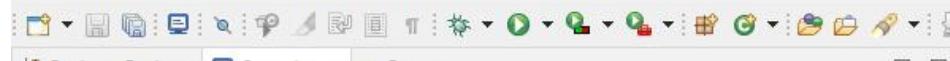
<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open  
enter any calendar year :2021  
2021 is not a leap year

```
1 package assessment;
2
3 //Accept year from user and validate if it is a leap year using ternary operator
4
5
6 import java.util.Scanner;
7
8 public class Practical_Assessment {
9     public static void main(String arg[])
10    {
11        long year;
12        Scanner sc=new Scanner(System.in);
13        System.out.print("enter any calendar year :");
14        year=sc.nextLong();
15        leapOrNot(year);
16    }
17    static void leapOrNot(long year)
18    {
19        if(year!=0)
20        {
21            if(year%400==0)
22                System.out.println(year+" is a leap year");
23            else if(year%100==0)
24                System.out.println(year+" is not a leap year");
25            else if(year%4==0)
26                System.out.println(year+" is a leap year");
27            else
28                System.out.println(year+" is not a leap year");
29        }
30        else
31            System.out.println("Year zero does not exist ");
32    }
33}
34
```



# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Package Explorer Console Progress

<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open

Enter marks for student:

95

Student passed:A+

```
7 //M > 80 is B+
8 //M > 70 is B
9 //M > 60 is C
10 //M > 50 is D
11 //M < 50 Fails.
12
13
14 import java.util.Scanner;
15
16 public class Practical_Assessment {
17     public static void main(String[] args) {
18         // TODO Auto-generated method stub
19         int marks;
20         Scanner sc=new Scanner(System.in);
21         System.out.println("Enter marks for student:");
22         marks=sc.nextInt();
23
24
25
26
27         if(marks>=95)
28             System.out.println("Student passed:A+");
29         else if(marks>=85 && marks<95)
30             System.out.println("Student Passed:A");
31         else if(marks>=80 && marks<85)
32             System.out.println("Student Passed:B+");
33         else if(marks>=70 && marks<80)
34             System.out.println("Student Passed:B");
35         else if(marks>=60 && marks<70)
36             System.out.println("Student Passed:C");
37         else if(marks>=50 && marks<60)
38             System.out.println("Student Passed:D");
39         else {
40             System.out.println("student failed");
41         }
42     }
43 }
44 }
```



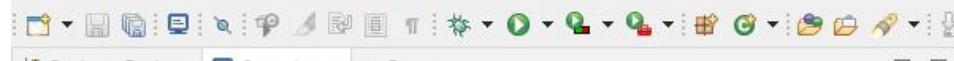
Type here to search



7:32 PM  
23°C  
12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open

Enter marks for student:

90

Student Passed:A

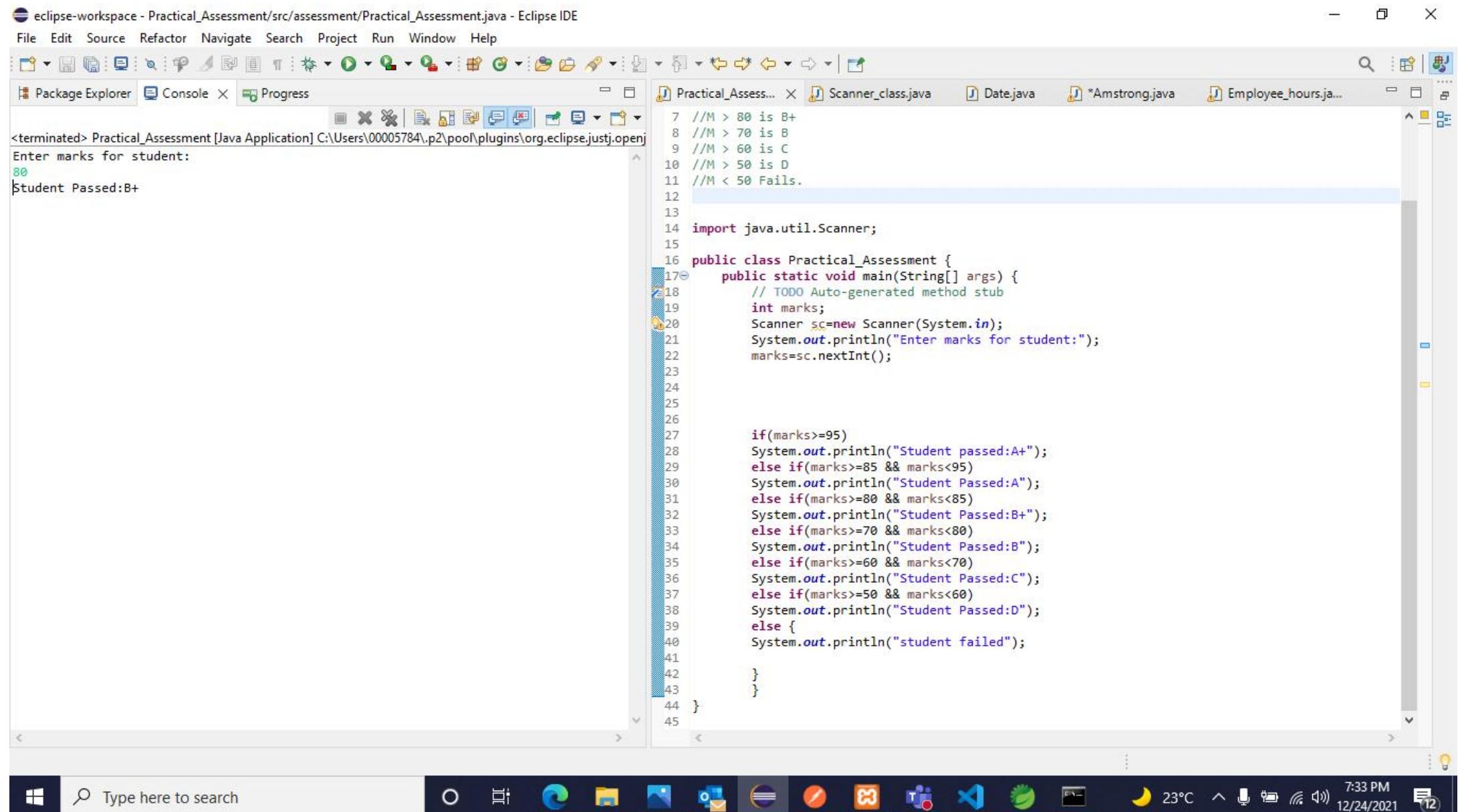
```
7 //M > 80 is B+
8 //M > 70 is B
9 //M > 60 is C
10 //M > 50 is D
11 //M < 50 Fails.
12
13
14 import java.util.Scanner;
15
16 public class Practical_Assessment {
17     public static void main(String[] args) {
18         // TODO Auto-generated method stub
19         int marks;
20         Scanner sc=new Scanner(System.in);
21         System.out.println("Enter marks for student:");
22         marks=sc.nextInt();
23
24
25
26
27         if(marks>=95)
28             System.out.println("Student passed:A+");
29         else if(marks>=85 && marks<95)
30             System.out.println("Student Passed:A");
31         else if(marks>=80 && marks<85)
32             System.out.println("Student Passed:B+");
33         else if(marks>=70 && marks<80)
34             System.out.println("Student Passed:B");
35         else if(marks>=60 && marks<70)
36             System.out.println("Student Passed:C");
37         else if(marks>=50 && marks<60)
38             System.out.println("Student Passed:D");
39         else {
40             System.out.println("student failed");
41         }
42     }
43 }
44 }
```



Type here to search



7:32 PM  
23°C 12/24/2021



# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project navigation.
- Left Sidebar:** Package Explorer, Console (containing the output of the run), and Progress.
- Central Area:** A code editor window titled "Practical\_Assess...". The code is a Java application that reads student marks and prints their grade based on a scale from 75 to 95. The code uses nested if-else statements and System.out.println() for output.
- Right Sidebar:** Shows other open files: Scanner\_class.java, Date.java, \*Amstrong.java, and Employee\_hours.java.

```
7 //M > 80 is B+
8 //M > 70 is B
9 //M > 60 is C
10 //M > 50 is D
11 //M < 50 Fails.
12
13
14 import java.util.Scanner;
15
16 public class Practical_Assessment {
17     public static void main(String[] args) {
18         // TODO Auto-generated method stub
19         int marks;
20         Scanner sc=new Scanner(System.in);
21         System.out.println("Enter marks for student:");
22         marks=sc.nextInt();
23
24
25
26
27         if(marks>=95)
28             System.out.println("Student passed:A+");
29         else if(marks>=85 && marks<95)
30             System.out.println("Student Passed:A");
31         else if(marks>=80 && marks<85)
32             System.out.println("Student Passed:B+");
33         else if(marks>=70 && marks<80)
34             System.out.println("Student Passed:B");
35         else if(marks>=60 && marks<70)
36             System.out.println("Student Passed:C");
37         else if(marks>=50 && marks<60)
38             System.out.println("Student Passed:D");
39         else {
40             System.out.println("student failed");
41         }
42     }
43 }
44
45
```



Type here to search



7:33 PM  
23°C  
12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

```
7 //M > 80 is A+
8 //M > 70 is B
9 //M > 60 is C
10 //M > 50 is D
11 //M < 50 Fails.
12
13
14 import java.util.Scanner;
15
16 public class Practical_Assessment {
17     public static void main(String[] args) {
18         // TODO Auto-generated method stub
19         int marks;
20         Scanner sc=new Scanner(System.in);
21         System.out.println("Enter marks for student:");
22         marks=sc.nextInt();
23
24
25
26
27         if(marks>=95)
28             System.out.println("Student passed:A+");
29         else if(marks>=85 && marks<95)
30             System.out.println("Student Passed:A");
31         else if(marks>=80 && marks<85)
32             System.out.println("Student Passed:B+");
33         else if(marks>=70 && marks<80)
34             System.out.println("Student Passed:B");
35         else if(marks>=60 && marks<70)
36             System.out.println("Student Passed:C");
37         else if(marks>=50 && marks<60)
38             System.out.println("Student Passed:D");
39         else {
40             System.out.println("student failed");
41         }
42     }
43 }
44 }
```



Type here to search



7:33 PM  
23°C  
12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project navigation.
- Views:** Package Explorer, Console, and Progress are visible in the top left.
- Console View:** Displays the output of the application run, showing the user input "Enter marks for student:" followed by "57" and the resulting output "Student Passed:D".
- Code Editor:** The main editor window contains the Java code for the `Practical_Assessment` class. The code uses a scanner to read student marks and prints a grade based on the following scale:
  - 95 >= marks >= 85: A+
  - 85 > marks >= 80: A
  - 80 > marks >= 70: B+
  - 70 > marks >= 60: B
  - 60 > marks >= 50: C
  - 50 > marks >= 0: D
  - marks < 50: Failed
- Project Explorer:** Shows files like `Scanner_class.java`, `Date.java`, `*Amstrong.java`, and `Employee_hours.java`.



Type here to search



7:33 PM  
23°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open

Enter marks for student:

49

student failed

```
7 //M > 80 is B+
8 //M > 70 is B
9 //M > 60 is C
10 //M > 50 is D
11 //M < 50 Fails.

12

13

14 import java.util.Scanner;
15

16 public class Practical_Assessment {
17     public static void main(String[] args) {
18         // TODO Auto-generated method stub
19         int marks;
20         Scanner sc=new Scanner(System.in);
21         System.out.println("Enter marks for student:");
22         marks=sc.nextInt();

23

24

25

26

27         if(marks>=95)
28             System.out.println("Student passed:A+");
29         else if(marks>=85 && marks<95)
30             System.out.println("Student Passed:A");
31         else if(marks>=80 && marks<85)
32             System.out.println("Student Passed:B+");
33         else if(marks>=70 && marks<80)
34             System.out.println("Student Passed:B");
35         else if(marks>=60 && marks<70)
36             System.out.println("Student Passed:C");
37         else if(marks>=50 && marks<60)
38             System.out.println("Student Passed:D");
39         else {
40             System.out.println("student failed");
41         }
42     }
43 }
44 }
```



Type here to search



# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



Package Explorer Console X Progress

<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open

Enter the amount

758

Total No of 500 note 1

Total No of 200 note 1

Total No of 50 note 1

Total No of 5 note 1

Total No of 2 note 1

Total No of 1 note 1

Total Number Of Notes Required is 5

```
1 package assessment;
2
3 //A banker has the following denominations to deliver cash. INR : 1,2,5,10,20,50,100,200,500.
4 //Accept a value from user and remit the sum with least number of notes.
5 //Test cases:
6 //In what denominations would a banker remit sum of Rs. 475, Rs. 530, Rs. 219
7 //Input: integer between 1 and 100,000
8 //Output: String describing the number of notes and currencies.
9
10
11 import java.util.Scanner;
12
13 public class Practical_Assessment {
14     public static void main(String[] args) {
15
16         int rs1 = 0, rs2 = 0, rs5 = 0, rs10 = 0, rs20 = 0, rs50 =
17         0, rs100 = 0, rs200 = 0, rs500 = 0;
18         Scanner input = new Scanner(System.in);
19         System.out.println("Enter the amount");
20         int amount = input.nextInt();
21         while (amount >= 500) {
22             rs500 = amount / 500;
23             amount = amount % 500;
24             System.out.println("Total No of 500 note " + rs500);
25         }
26         while (amount >= 200) {
27             rs200 = amount / 200;
28             amount = amount % 200;
29             System.out.println("Total No of 200 note " + rs200);
30         }
31         while (amount >= 100) {
32             rs100 = amount / 100;
33             amount = amount % 100;
34             System.out.println("Total No of 100 note " + rs100);
35         }
36         while (amount >= 50) {
37             rs50 = amount / 50;
38             amount = amount % 50;
39             System.out.println("Total No of 50 note " + rs50);
```



Type here to search



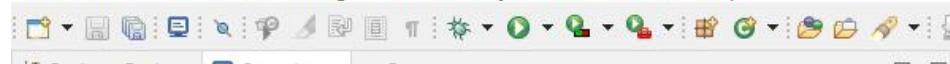
7:38 PM

23°C

12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open

Enter the number to display month

3

March

```
1 package assessment;
2
3 //Using Switch statement:
4 //Print the month of the year in literals, based on the numeric value entered.
5 //UserInput: 1
6 //Output: January
7 //UserInput: 9
8 //Output: September
9
10
11 import java.util.Scanner;
12
13 public class Practical_Assessment {
14     public static void main(String[] args) {
15
16         Scanner s=new Scanner(System.in);
17         System.out.println("Enter the number to display month");
18         int month = s.nextInt();
19         switch (month) {
20             case 1:
21                 System.out.println("January");
22                 break;
23             case 2:
24                 System.out.println("February");
25                 break;
26             case 3:
27                 System.out.println("March");
28                 break;
29             case 4:
30                 System.out.println("April");
31                 break;
32             case 5:
33                 System.out.println("May");
34                 break;
35             case 6:
36                 System.out.println("June");
37                 break;
38             case 7:
39                 System.out.println("July");
```



Type here to search



# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- View Bar:** Package Explorer, Console (terminated), Progress.
- Code Editor:** The main editor window displays the Java code for the `Practical_Assessment` class. The code uses a switch statement to map month numbers to their corresponding names.

```
1 package assessment;
2
3 //Using Switch statement:
4 //Print the month of the year in literals, based on the numeric value entered.
5 //UserInput: 1
6 //Output: January
7 //UserInput: 9
8 //Output: September
9
10 import java.util.Scanner;
11
12
13 public class Practical_Assessment {
14     public static void main(String[] args) {
15
16         Scanner s=new Scanner(System.in);
17         System.out.println("Enter the number to display month");
18         int month = s.nextInt();
19         switch (month) {
20             case 1:
21                 System.out.println("January");
22                 break;
23             case 2:
24                 System.out.println("February");
25                 break;
26             case 3:
27                 System.out.println("March");
28                 break;
29             case 4:
30                 System.out.println("April");
31                 break;
32             case 5:
33                 System.out.println("May");
34                 break;
35             case 6:
36                 System.out.println("June");
37                 break;
38             case 7:
39                 System.out.println("July");
```

- Console View:** Shows the output of the application: "Enter the number to display month" followed by "12" and "December".
- Bottom Status Bar:** Shows the Windows taskbar with various pinned icons and system status information (7:45 PM, 23°C, 12/24/2021).



Type here to search



7:45 PM  
23°C  
12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- View Bar:** Package Explorer, Console (terminated), Progress.
- Code Editor:** The main editor window displays the Java code for the `Practical_Assessment` class. The code uses a switch statement to map month numbers to their names.

```
1 package assessment;
2
3 //Using Switch statement:
4 //Print the month of the year in literals, based on the numeric value entered.
5 //UserInput: 1
6 //Output: January
7 //UserInput: 9
8 //Output: September
9
10 import java.util.Scanner;
11
12
13 public class Practical_Assessment {
14     public static void main(String[] args) {
15
16         Scanner s=new Scanner(System.in);
17         System.out.println("Enter the number to display month");
18         int month = s.nextInt();
19         switch (month) {
20             case 1:
21                 System.out.println("January");
22                 break;
23             case 2:
24                 System.out.println("February");
25                 break;
26             case 3:
27                 System.out.println("March");
28                 break;
29             case 4:
30                 System.out.println("April");
31                 break;
32             case 5:
33                 System.out.println("May");
34                 break;
35             case 6:
36                 System.out.println("June");
37                 break;
38             case 7:
39                 System.out.println("July");
```

- Console View:** Shows the output of the application. It prints "Enter the number to display month", then "14", and then "Invalid input".



Type here to search



7:45 PM  
23°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Date.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Date.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- Left Sidebar:** Package Explorer, Console (terminated output: Enter Date (dd/MM/yyyy) : 24/12/2021, Valid Date), Progress.
- Central Area:** Text editor showing Java code for validating dates. The code uses Scanner to read a date string and then checks if it's a valid date by comparing month and day against an array of days in each month.
- Right Sidebar:** Standard Eclipse sidebar icons.

```
1 package assessment;
2
3 import java.util.Scanner;
4
5 //The following needs String class.
6 //Using Switch:
7 //1. Accept a past date from user and check if it is a valid date.
8 //Do not use 'SimpleDateFormat' class from java.
9 //Example
10 //Input:
11 //Enter Date(DD/MM/YYYY): 12/12/2009
12 //This is a valid date.
13 //Input:
14 //Enter Date(DD/MM/YYYY): 29/02/2009
15 //This is invalid date.
16
17
18 public class Date {
19     public static void main(String []args)
20     {
21         Scanner sc=new Scanner(System.in);
22         int days[]={31,28,31,30,31,30,31,31,30,31,30,31};
23         String input;
24         System.out.print("Enter Date (dd/MM/yyyy) : ");
25         input=sc.nextLine();
26         int d=Integer.parseInt(input.substring(0,2));
27         int m=Integer.parseInt(input.substring(3,5));
28         int y=Integer.parseInt(input.substring(6));
29         if(y>1900 && y<=9999)
30         {
31             if(((y % 4 == 0) && (y % 100 != 0)) || (y % 400 == 0))
32             {
33                 days[1]++;
34             }
35             if(m>0 && m<=12)
36             {
37                 if(d>0 && d< days[m-1])
38                 {
39                     System.out.print("Valid Date");
40                 }
41             }
42         }
43     }
44 }
```



Type here to search



7:54 PM  
22°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Date.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Date.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- Left Sidebar:** Package Explorer, Console (with output: <terminated> Date [Java Application] C:\Users\00005784.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.fu), Progress.
- Central Area:** Text editor showing Java code for validating dates. The code uses Scanner to read a date string and then checks if it's valid by comparing month and day against an array of days in a year. It handles leap years and invalid inputs.
- Right Sidebar:** Standard Eclipse sidebar icons.

```
1 package assessment;
2
3 import java.util.Scanner;
4
5 //The following needs String class.
6 //Using Switch:
7 //1. Accept a past date from user and check if it is a valid date.
8 //Do not use 'SimpleDateFormat' class from java.
9 //Example
10 //Input:
11 //Enter Date(DD/MM/YYYY): 12/12/2009
12 //This is a valid date.
13 //Input:
14 //Enter Date(DD/MM/YYYY): 29/02/2009
15 //This is invalid date.
16
17
18 public class Date {
19     public static void main(String []args)
20     {
21         Scanner sc=new Scanner(System.in);
22         int days[]={31,28,31,30,31,30,31,31,30,31,30,31};
23         String input;
24         System.out.print("Enter Date (dd/MM/yyyy) : ");
25         input=sc.nextLine();
26         int d=Integer.parseInt(input.substring(0,2));
27         int m=Integer.parseInt(input.substring(3,5));
28         int y=Integer.parseInt(input.substring(6));
29         if(y>1900 && y<=9999)
30         {
31             if(((y % 4 == 0) && (y % 100 != 0)) || (y % 400 == 0))
32             {
33                 days[1]++;
34             }
35             if(m>0 && m<=12)
36             {
37                 if(d>0 && d< days[m-1])
38                 {
39                     System.out.print("Valid Date");
40                 }
41             }
42         }
43     }
44 }
```



Type here to search

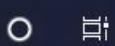


7:55 PM  
22°C 12/24/2021

```
J *Amstrong.java X
1 package assessment;
2
3 import java.util.Scanner;
4
5
6 //1. Can there be return type for a constructor?
7 //Ans: No
8
9 //2. Can we define a non default constructor without define default constructor
10 //Ans : Yes
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
```



Type here to search



7:58 PM  
22°C  
12/24/2021



```
J *Amstrong.java X
4
5
6 //what is wrong with this?
7 //for (int i = 1; i <= 1000; i++) {
8 // int sum = 0;
9 // sum = sum + i;
10 //}
11 //System.out.println("The sum is " + sum);
12 //
13 //
14 // Ans : sum cannot be accessed outside the for loop because it is initialised inside the
15 // for loop
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
```

Type here to search    8:01 PM  
12/24/2021

eclipse-workspace - Practical\_Assessment/src/assessment/Employee\_hours.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Employee\_hours.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- View Bar:** Package Explorer, Console, Progress.
- Code Editor:** Displays the Java code for `Employee_hours.java`. The code reads a number of hours from the user, calculates payment based on 250 per hour for up to 8 hours, and prints a message for overtime.

```
1 package assessment;
2
3 import java.util.Scanner;
4
5 //Gather the number of hours employee works, pay only the eight hours he works at the rate of Rs.
6 //per hour.
7 //It is a company policy not to pay over time.
8
9 public class Employee_hours {
10    public static void main(String a[]){
11        Scanner s = new Scanner(System.in);
12        System.out.print("Enter the number of hours you worked : ");
13        int hour=s.nextInt();
14        if(hour<=8) {
15            System.out.print("Payment for "+hour+" hours of work is Rs."+ (250*hour));
16        }
17        else {
18            System.out.println("Payment for "+hour+" hours of work is Rs."+(250*8));
19            System.out.println("No payment for over time");
20        }
21    }
22}
23
```

- Console View:** Shows the output of the application: "Enter the number of hours you worked : 4" and "Payment for 4 hours of work is Rs.1000".
- Bottom Status Bar:** Writable, Smart Insert, 23:1:706.



Type here to search



8:02 PM  
22°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Employee\_hours.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

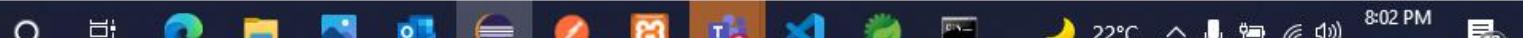
- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project navigation.
- ActionBar:** Shows tabs for "Package Explorer", "Console X", and "Progress".
- Console View:** Displays the output of a Java application named "Employee\_hours". The output is:

```
<terminated> Employee_hours [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\openjdk.h
Enter the number of hours you worked : 10
Payment for 10 hours of work is Rs.2000
No payment for over time
```
- Code Editor:** The "Employee\_hours.java" file is open in the editor. The code is as follows:

```
1 package assessment;
2
3 import java.util.Scanner;
4
5 //Gather the number of hours employee works, pay only the eight hours he works at the rate of Rs.
6 //per hour.
7 //It is a company policy not to pay over time.
8
9 public class Employee_hours {
10    public static void main(String a[]){
11        Scanner s = new Scanner(System.in);
12        System.out.print("Enter the number of hours you worked : ");
13        int hour=s.nextInt();
14        if(hour<=8) {
15            System.out.print("Payment for "+hour+" hours of work is Rs."+ (250*hour));
16        }
17        else {
18            System.out.println("Payment for "+hour+" hours of work is Rs."+(250*8));
19            System.out.println("No payment for over time");
20        }
21    }
22}
23
```



Type here to search



8:02 PM  
22°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Employee\_hours.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project navigation.
- ActionBar:** Shows tabs for "Package Explorer", "Console X", and "Progress".
- Console View:** Displays the output of a Java application named "Employee\_hours". The output is:

```
<terminated> Employee_hours [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.openjdk.h
Enter the base :
2
Enter the exponent
3
the raised power is:8
```
- Code Editor:** The "Employee\_hours.java" file is open. The code defines a class "Employee\_hours" with a main method that reads a base and exponent from the user and prints the result. It also contains a recursive power method.
- Project Bar:** Shows other files in the project: "Practical\_Assessme...", "Scanner\_class.java", "Date.java", "Employee\_hours.java X", and "\*Credit.java".

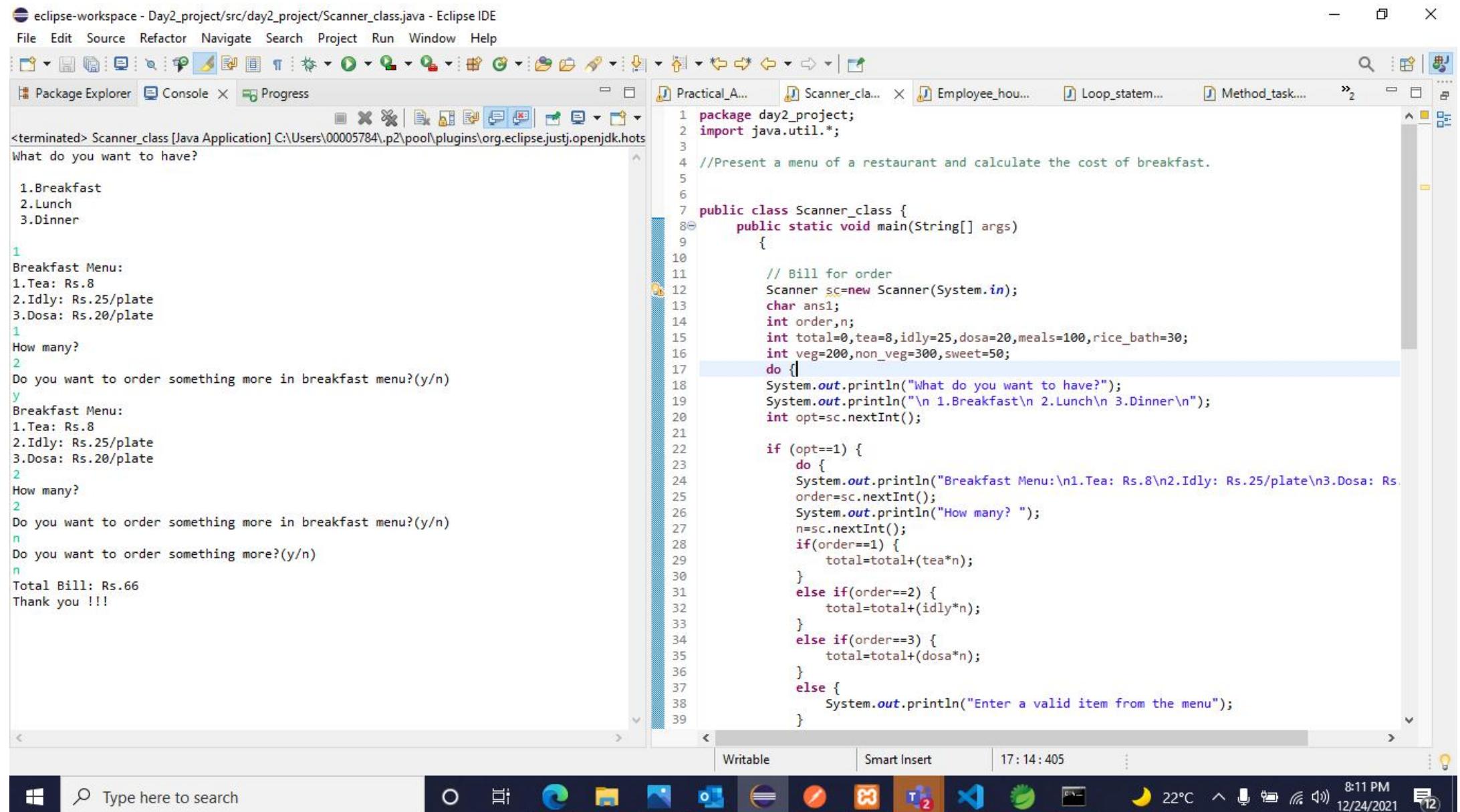
```
1 package assessment;
2
3 import java.util.Scanner;
4 //
5 //Accept a base b, and an exponent e.
6 //write a function power(b,e) that returns b raised e. Print the solution.
7
8 public class Employee_hours {
9     public static void main(String a[] ){
10        Scanner sc=new Scanner(System.in);
11        System.out.println("Enter the base :");
12        int b=sc.nextInt();
13        System.out.println("Enter the exponent");
14        int e=sc.nextInt();
15        System.out.println("the raised power is:"+power(b,e));
16    }
17    public static int power(int b,int e) {
18        if(e==0)
19        {
20            return 1;
21        }
22        else
23        {
24            return b*power(b,e-1);
25        }
26    }
27
28}
```



Type here to search



8:08 PM  
22°C 12/24/2021



# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project navigation.
- Views:** Package Explorer, Console, and Progress are visible in the top bar.
- Console View:** Shows the output of the Java application: "Armstrong numbers from 1 to 500 are" followed by the numbers 1, 153, 370, 371, and 407.
- Code Editor:** The main editor window displays the Java code for the `Practical_Assessment` class. The code uses a while loop to iterate through numbers 1 to 500, calculates the sum of cubes of digits for each number, and prints it if it equals the original number (Armstrong number).
- Project Bar:** Shows other open projects like `Scanner_cla...`, `Employee_hou...`, `Loop_statem...`, and `Method_task...`.



Type here to search

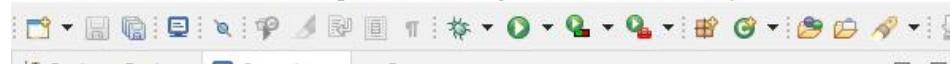


8:13 PM  
22°C 12/24/2021



<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open  
Enter the number  
371  
371 is a Amstrong number

```
1 package assessment;
2
3 //An Armstrong number is a n-digit number that is equal to the sum of nth power of its digits.
4 //Example all single digit number are Armstrong numbers
5 //3 to power 1 is 3
6 //371 = 3 Power 3 + 7 power 3 + 1 power 3 = 371.
7 //Changes:Modify the code to verify if a given number is Armstrong number
8
9
10 import java.util.Scanner;
11
12 public class Practical_Assessment {
13     public static void main(String[] args) {
14         int i,p,arm,n;
15         Scanner s=new Scanner(System.in);
16         System.out.println("Enter the number");
17         i=s.nextInt();
18         n=i;
19         arm=0;
20         while(n>0)
21         {
22             p=n%10;
23             arm=arm+(p*p*p);
24             n=n/10;
25         }
26         if(arm==i) {
27             System.out.println( +i + " is a Amstrong number");
28         }
29
30         else {
31             System.out.println( +i + " is not a Amstrong number ");
32         }
33         i++;
34
35
36
37
38     }
39 }
```



<terminated> Practical\_Assessment [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.core\open  
Enter the number  
500  
500 is not a Amstrong number

```
1 package assessment;
2
3 //An Armstrong number is a n-digit number that is equal to the sum of nth power of its digits.
4 //Example all single digit number are Armstrong numbers
5 //3 to power 1 is 3
6 //371 = 3 Power 3 + 7 power 3 + 1 power 3 = 371.
7 //Changes:Modify the code to verify if a given number is Armstrong number
8
9
10 import java.util.Scanner;
11
12 public class Practical_Assessment {
13     public static void main(String[] args) {
14         int i,p,arm,n;
15         Scanner s=new Scanner(System.in);
16         System.out.println("Enter the number");
17         i=s.nextInt();
18         n=i;
19         arm=0;
20         while(n>0)
21         {
22             p=n%10;
23             arm=arm+(p*p*p);
24             n=n/10;
25         }
26         if(arm==i) {
27             System.out.println( +i + " is a Amstrong number");
28         }
29
30         else {
31             System.out.println( +i + " is not a Amstrong number ");
32         }
33         i++;
34     }
35
36
37
38
39 }
```

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project management.
- Views:** Package Explorer, Console, and Progress are visible in the top bar.
- Console View:** Displays the output of the application: "Enter a Number:" followed by the user input "9". Below it, the program prints all factors of numbers from 1 to 9.
- Code Editor:** The main window displays the Java code for the `Practical_Assessment` class. The code uses nested loops to count factors for each number from 1 to `N`.
- Code Snippet:** The code is annotated with comments explaining its purpose and logic.

```
1 package assessment;
2
3 //Count the number of factors for every number from 1 to N.
4 //Example:
5 //N = 9
6 //Factors 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
7
8
9 import java.util.Scanner;
10
11 public class Practical_Assessment {
12     public static void main(String[] args) {
13         Scanner input = new Scanner(System.in);
14         System.out.println("Enter a Number:");
15         int num = input.nextInt();
16         for (int i = 1; i <= num; i++) {
17             int count = 0;
18             for (int j = 1; j <= i; j++) {
19                 if (i % j == 0)
20                     count++;
21             }
22             System.out.println("Factor of " + i + " is " + count);
23         }
24     }
25
26
27 }
```



Type here to search



8:16 PM  
22°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Employee\_hours.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Employee\_hours.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- View Bar:** Package Explorer, Console, Progress.
- Code Editor:** Displays Java code for generating a diamond pattern. The code uses nested loops and System.out.print() to print asterisks (\*).

```
1 package assessment;
2
3 import java.util.Scanner;
4
5 //1. Print the following pattern on the screen.
6 /**
7 /**
8 /**
9 /**
10 /**
11
12 public class Employee_hours {
13     public static void main(String args[])
14     {
15         int i, j;
16         int r = 5;
17         // outer loop
18         for(i=0;i<r;i++)
19         {
20             //inner loop
21             for(j=0;j<=i; j++)
22             {
23                 System.out.print("*");
24             }
25             System.out.println();
26         }
27     }
28 }
```

- Bottom Status Bar:** Writable, Smart Insert, 27:6:388.



Type here to search



8:22 PM  
22°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Employee\_hours.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Employee\_hours.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- View Bar:** Package Explorer, Console, Progress.
- Code Editor:** Displays Java code for generating a diamond pattern. The code includes imports for `java.util.Scanner` and a main method that uses nested loops to print asterisks (\*) to the console.

```
1 package assessment;
2
3 import java.util.Scanner;
4
5 //2. Accept height and width and print the pattern with a '**'
6 //Ex for height = 5 , width = 5,
7 /**
8 /**
9 /**
10 /**
11 /**
12
13 public class Employee_hours {
14     public static void main(String[] args) {
15         // TODO Auto-generated method stub
16         int i,j;
17         for(i=1;i<=5;i++) {
18             for(j=1;j<=5;j++) {
19                 System.out.print("**");
20             }
21             System.out.println();
22         }
23     }
24 }
25
```

- Bottom Status Bar:** Writable, Smart Insert, 25 : 1 : 456.



Type here to search



8:23 PM

22°C

12/24/2021

eclipse-workspace - Practical\_Assessment/src/assessment/Employee\_hours.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project management.
- ActionBar:** Shows tabs for "Package Explorer", "Console X", and "Progress".
- Console View:** Displays the output of the application: "`<terminated> Employee_hours [Java Application] C:\Users\00005784.p2\pool\plugins\org.eclipse.justj.openjdk.h`". Below it, the application's output is shown: "Fibonacci Series till 10 terms:  
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,".
- Code Editor:** The main workspace displays the Java code for the `Employee_hours` class. The code prints the first 10 terms of the Fibonacci series using a for loop.
- Right-hand Side:** Shows the Java editor's right margin with status indicators like "Writable" and "Smart Insert", and a timestamp "5:33:84".

```
1 package assessment;
2
3 import java.util.Scanner;
4
5 //Fibonacci series till 10 terms
6
7 public class Employee_hours {
8     public static void main(String[] args) {
9
10         int n = 10, firstTerm = 0, secondTerm = 1;
11         System.out.println("Fibonacci Series till " + n + " terms:");
12         for (int i = 1; i <= n; ++i) {
13             System.out.print(firstTerm + ", ");
14             // compute the next term
15             int nextTerm = firstTerm + secondTerm;
16             firstTerm = secondTerm;
17             secondTerm = nextTerm;
18
19
20
21         }
22     }
23 }
```



Type here to search



8:24 PM  
22°C 12/24/2021

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

- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project navigation.
- ActionBar:** Shows tabs for "Package Explorer", "Console X", and "Progress".
- Console View:** Displays the output of the Java application. It shows the user input "Enter the number : 145" and the program's response "145 is a strong number".
- Code Editor:** The main workspace displays the Java code for the `Employee\_hours` class. The code uses a `Scanner` to read an integer from the user. It then calculates the sum of the factorials of each digit of the number. If this sum equals the original number, it is considered a "strong number". The code includes comments explaining the logic.

```
3 import java.util.Scanner;
4
5 //Definition of strong number:
6 //The sum of the factorials of digits of a number should add upto the number itself.
7 //Example
8 //145 => 1! + 4! + 5!
9
10 public class Employee_hours {
11     public static void main(String[] args) {
12         int n,i;
13         int fact,ldig;
14         Scanner s = new Scanner(System.in);
15         System.out.print("Enter the number : ");
16         n = s.nextInt();
17         int total = 0;
18         int temp = n;
19         while(n != 0)
20         {
21             i = 1;
22             fact = 1;
23             ldig = n % 10;
24             while(i <= ldig)
25             {
26                 fact = fact * i;
27                 i++;
28             }
29             total = total + fact;
30             n = n / 10;
31         }
32         if(total == temp)
33             System.out.println(temp + " is a strong number\n");
34         else
35             System.out.println(temp + " is not a strong number\n");
36         System.out.println();
37
38     }
39 }
40 }
41 }
```



Type here to search

8:26 PM  
22°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Employee\_hours.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project navigation.
- Left Sidebar:** Package Explorer, Console (containing the output of the run), and Progress.
- Top Status Bar:** Shows the current file path: \*Practical\_A... Employee\_hou... X Loop\_statem... Method\_task...\*.
- Code Editor:** Displays Java code for a class named Employee\_hours. The code uses a Scanner to read a number from the user and calculates its total factorial of digits. It then compares this total with the original number to determine if it's a strong number. The code includes comments explaining the logic.

```
3 import java.util.Scanner;
4
5 //Definition of strong number:
6 //The sum of the factorials of digits of a number should add upto the number itself.
7 //Example
8 //145 => 1! + 4! + 5!
9
10 public class Employee_hours {
11     public static void main(String[] args) {
12         int n,i;
13         int fact,ldig;
14         Scanner s = new Scanner(System.in);
15         System.out.print("Enter the number : " );
16         n = s.nextInt();
17         int total = 0;
18         int temp = n;
19         while(n != 0)
20         {
21             i = 1;
22             fact = 1;
23             ldig = n % 10;
24             while(i <= ldig)
25             {
26                 fact = fact * i;
27                 i++;
28             }
29             total = total + fact;
30             n = n / 10;
31         }
32         if(total == temp)
33             System.out.println(temp + " is a strong number\n");
34         else
35             System.out.println(temp + " is not a strong number\n");
36         System.out.println();
37
38     }
39 }
40 }
```

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Employee\_hours.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- View Bar:** Package Explorer, Console, Progress.
- Console View:** Displays the output: <terminated> Employee\_hours [Java Application] C:\Users\00005784.p2\pool\plugins\org.eclipse.justj.openjdk.h... Yes
- Code Editor:** The main window displays the Java code for the Employee\_hours class. The code uses a HashMap to count characters in the first string and then checks if any character from the second string exists in the map.

```
2
3 import java.util.HashMap;
4 import java.util.Map;
5 import java.util.Scanner;
6
7 //Find if there is any character common between the two strings.
8 //If there is a common character return YES else return "NO"
9 //UserInputs:
10 //1. Accept two strings from user.
11 //Output:
12 //Display "YES"
13
14 public class Employee_hours {
15
16     static boolean check(String s1, String s2)
17     {
18         Map<Character, Integer> mp = new HashMap<>();
19         for (int i = 0; i < s1.length(); i++)
20         {
21             mp.put(s1.charAt(i), mp.get(s1.charAt(i)) == null ? 1 : mp.get(s1.charAt(i)) + 1);
22         }
23         for (int i = 0; i < s2.length(); i++)
24         {
25             if (mp.get(s2.charAt(i)) > 0)
26             {
27                 return true;
28             }
29         }
30         return false;
31     }
32     public static void main(String[] args)
33     {
34         String s1 = "arisglobal", s2 = "aris";
35         boolean yes_or_no = check(s1, s2);
36
37         if (yes_or_no == true)
38         {
39             System.out.println("Yes");
40         }
41     }
42 }
```



Type here to search

8:29 PM  
21°C  
12/24/2021

The screenshot shows the Eclipse IDE interface with a Java application running. The code in the editor reads a string from standard input, converts it to a character array, and prints each character to standard output.

```
1 package assessment;
2
3 import java.util.HashMap;
4 import java.util.Map;
5 import java.util.Scanner;
6
7 //Convert string to char array
8
9
10 public class Employee_hours {
11
12     public static void main (String arg[])
13     {
14         String str;
15         Scanner sc=new Scanner(System.in);
16         System.out.println("enter the String:");
17         str=sc.next();
18         char[] c=new char[str.length()];
19         for(int i=0;i<str.length();i++)
20         {
21             c[i]=str.charAt(i);
22         }
23         System.out.println("The array is:");
24
25
26
27         for(char a: c)
28         {
29             System.out.println(a);
30
31
32         }
33     }
34 }
35
36
```



Type here to search

8:30 PM  
21°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Credit.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Credit.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Standard Eclipse toolbar icons.
- View Bar:** Package Explorer, Console (terminated output: Credit [Java Application] C:\Users\00005784\.p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.f...), Progress.
- Code Editor:** The current file is Credit.java, located in the package assessment. The code implements a method to validate credit card numbers based on Luhn's algorithm.

```
1 package assessment;
2
3 public class Credit {
4     public static void main(String[] args) {
5         long number = 5196081888500645L;
6
7
8
9         System.out.println(number + " is " +
10            (isValid(number) ? "valid" : "invalid"));
11    }
12    public static boolean isValid(long number)
13    {
14        return (getSize(number) >= 13 &&
15        getSize(number) <= 16) ||
16        prefixMatched(number, 4) ||
17        prefixMatched(number, 5) ||
18        prefixMatched(number, 37) ||
19        prefixMatched(number, 6)) &&
20        ((sumOfDoubleEvenPlace(number) +
21        sumOfOddPlace(number)) % 10 == 0);
22    }
23    public static int sumOfDoubleEvenPlace(long number)
24    {
25        int sum = 0;
26        String num = number + "";
27        for (int i = getSize(number) - 2; i >= 0; i -= 2)
28            sum += getDigit(Integer.parseInt(num.charAt(i) + "") * 2);
29
30
31
32        return sum;
33    }
34    public static int getDigit(int number)
35    {
36        if (number < 9)
37            return number;
38        return number / 10 + number % 10;
39    }
}
```




Type here to search



8:35 PM  
21°C 12/24/2021

# eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

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

- Title Bar:** eclipse-workspace - Practical\_Assessment/src/assessment/Practical\_Assessment.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Toolbar:** Includes icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, and others.
- ActionBar:** Includes icons for Run, Stop, Refresh, and others.
- Left Sidebar:** Package Explorer, Console, Progress.
- Right Sidebar:** Shows tabs for other Java files: Practical\_A..., Employee\_hou..., Credit.java, Loop\_statem..., Method\_task....
- Code Editor:** Displays Java code for calculating the Highest Common Factor (HCF) of two numbers, 32 and 58. The code uses dynamic programming to store intermediate results in a 2D array dp.

```
1 package assessment;
2
3 //HCF: Highest Common Factor.
4 //The HCF of two numbers is the greatest number that divides the two numbers.
5 //Example:
6 //HCF of (14, 42) is 14.
7 //Factors of 14 -> 2, 7, 14
8 //Factors of 42 -> 2, 3, 6, 7, 14, 21
9
10 import java.util.Scanner;
11
12 public class Practical_Assessment {
13     // public class HCF {
14     static int [][]dp = new int[1001][1001];
15     static int gcd(int a, int b)
16     {
17         if (a == 0)
18             return b;
19         if (b == 0)
20             return a;
21         if (a == b)
22             return a;
23         if(dp[a][b] != -1)
24             return dp[a][b];
25         if (a > b)
26             dp[a][b] = gcd(a-b, b);
27         else
28             dp[a][b] = gcd(a, b-a);
29         return dp[a][b];
30     }
31
32     public static void main(String[] args)
33     {
34         for(int i = 0; i < 1001; i++)
35             for(int j = 0; j < 1001; j++)
36                 dp[i][j] = -1;
37     }
38
39     int a = 32, b = 58;
```

- Bottom Status Bar:** Writable, Smart Insert, 39 : 27 : 805.
- Taskbar:** Shows various pinned application icons.
- System Tray:** Shows the date (12/24/2021), time (8:39 PM), battery level (12%), and system status.



Type here to search



8:39 PM  
21°C  
12/24/2021