

**CS4001 - Programming**  
**Syntax and Semantics of Programming Language**  
**Week 03 : Java**

**Question 1:**

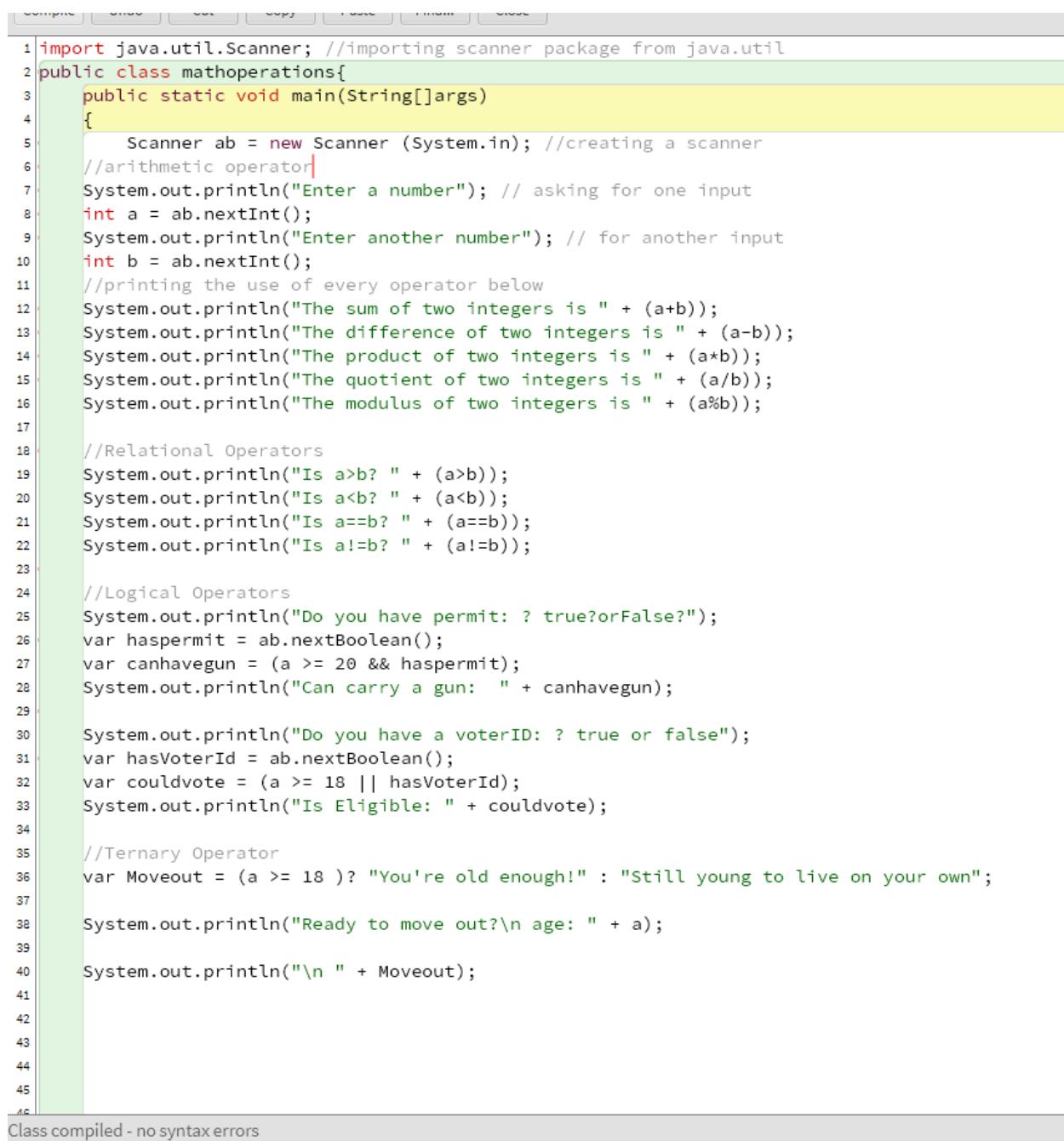
Create "MathOperations.java" with all operator types

**Test case**

Test case

Objective	To compile and execute MathOperations.java and using all the different types of operators.
Action	<ul style="list-style-type: none"><li>• Bluej was opened</li><li>• MathOperations.java class was formed</li><li>• Used all the operators after taking two int inputs from the user.</li><li>• Compiled the program with necessary comments alongside for readability.</li><li>• Successfully executed the program.</li></ul>
ExpectedResult	No compile-time errors and displayed the output.
ActualResult	Program ran smoothly as expected without any errors
Conclusion	Successful test results obtained

## Written and Compiled Code:



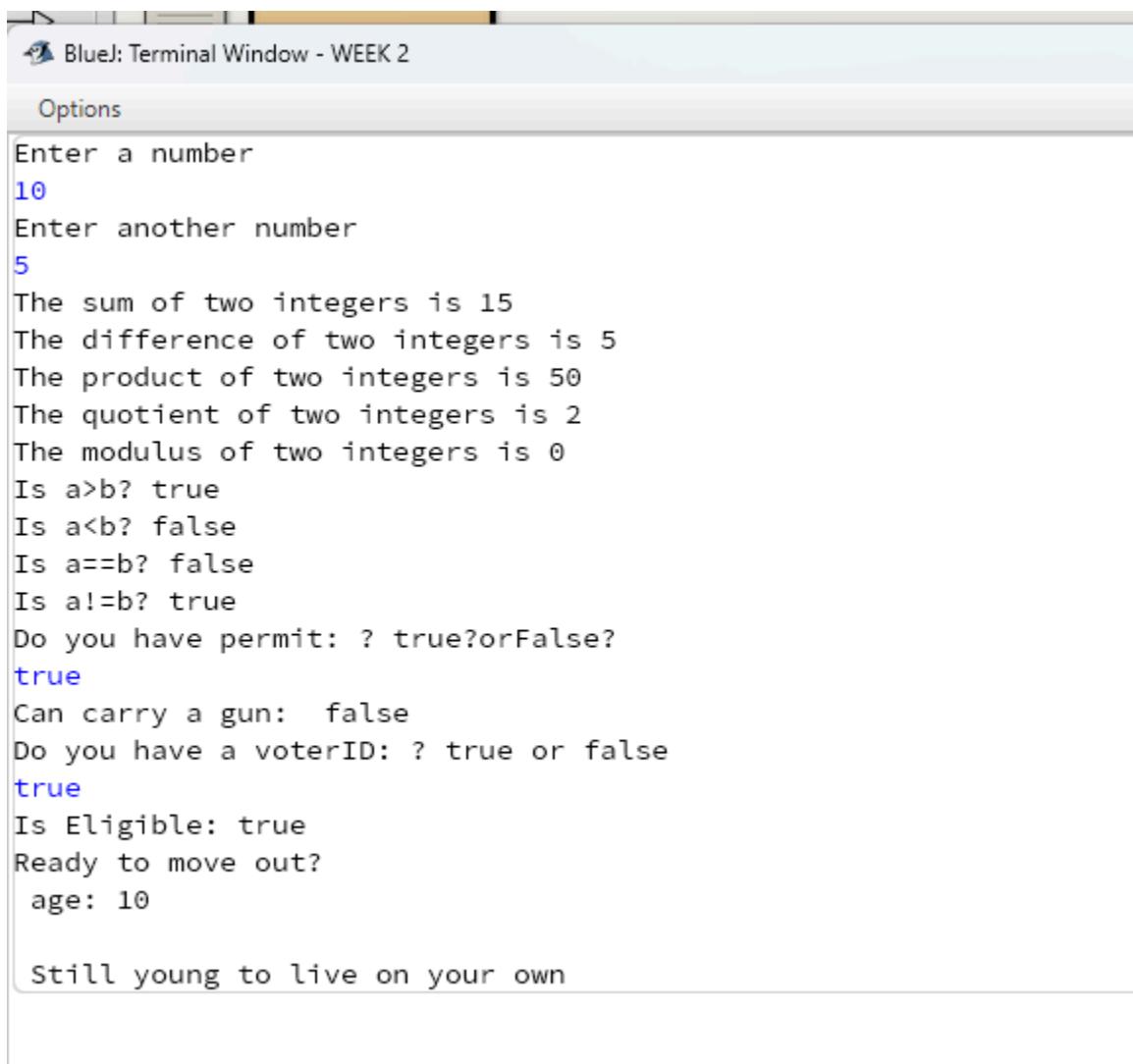
```

1 import java.util.Scanner; //importing scanner package from java.util
2 public class mathoperations{
3     public static void main(String[]args)
4     {
5         Scanner ab = new Scanner (System.in); //creating a scanner
6         //arithmetic operator
7         System.out.println("Enter a number"); // asking for one input
8         int a = ab.nextInt();
9         System.out.println("Enter another number"); // for another input
10        int b = ab.nextInt();
11        //printing the use of every operator below
12        System.out.println("The sum of two integers is " + (a+b));
13        System.out.println("The difference of two integers is " + (a-b));
14        System.out.println("The product of two integers is " + (a*b));
15        System.out.println("The quotient of two integers is " + (a/b));
16        System.out.println("The modulus of two integers is " + (a%b));
17
18        //Relational Operators
19        System.out.println("Is a>b? " + (a>b));
20        System.out.println("Is a<b? " + (a<b));
21        System.out.println("Is a==b? " + (a==b));
22        System.out.println("Is a!=b? " + (a!=b));
23
24        //Logical Operators
25        System.out.println("Do you have permit: ? true?orFalse?");
26        var haspermit = ab.nextBoolean();
27        var canhavegun = (a >= 20 && haspermit);
28        System.out.println("Can carry a gun: " + canhavegun);
29
30        System.out.println("Do you have a voterID: ? true or false");
31        var hasVoterId = ab.nextBoolean();
32        var couldvote = (a >= 18 || hasVoterId);
33        System.out.println("Is Eligible: " + couldvote);
34
35        //Ternary Operator
36        var Moveout = (a >= 18 )? "You're old enough!" : "Still young to live on your own";
37
38        System.out.println("Ready to move out?\n age: " + a);
39
40        System.out.println("\n " + Moveout);
41
42
43
44
45
46

```

Class compiled - no syntax errors

### Actual Output:



The screenshot shows a BlueJ terminal window titled "BlueJ: Terminal Window - WEEK 2". The window contains the following text output:

```
Enter a number
10
Enter another number
5
The sum of two integers is 15
The difference of two integers is 5
The product of two integers is 50
The quotient of two integers is 2
The modulus of two integers is 0
Is a>b? true
Is a<b? false
Is a==b? false
Is a!=b? true
Do you have permit: ? true?orFalse?
true
Can carry a gun: false
Do you have a voterID: ? true or false
true
Is Eligible: true
Ready to move out?
age: 10

Still young to live on your own
```

## Question 2: GradeEvaluator.java

Create a program that:

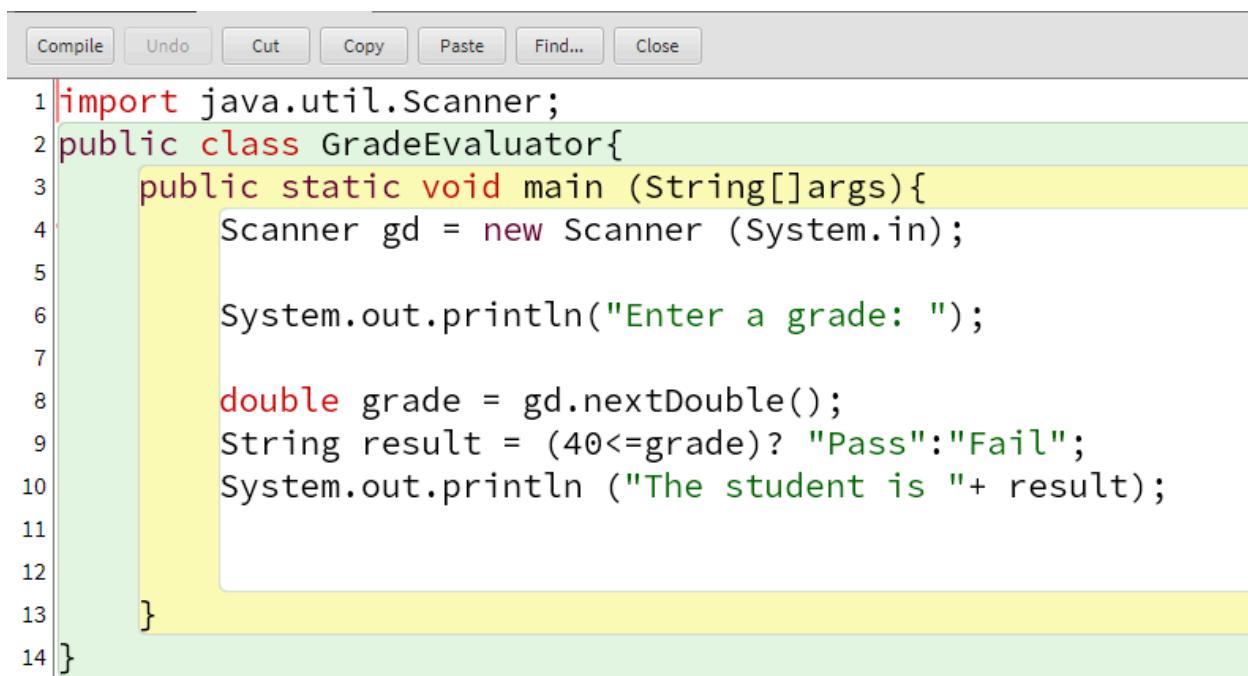
- Takes a numeric grade as input
- Uses the ternary operator to assign:
- "Pass" if  $\text{grade} \geq 40$
- "Fail" if  $\text{grade} < 40$

### Test case

#### Test case

Objective	To compile and execute GradeEvaluator.java and using two variables and boolean operator
Action	<ul style="list-style-type: none"><li>• Bluej was opened</li><li>• GradeEvaluator.java class was formed</li><li>• Took one input(grade) from the user to check Pass or Fail.</li><li>• Compiled the program with necessary comments alongside for readability.</li><li>• Successfully executed the program.</li></ul>
ExpectedResult	No compile-time errors and displayed the output.
ActualResult	Program ran smoothly as expected without any errors
Conclusion	Successful test results obtained

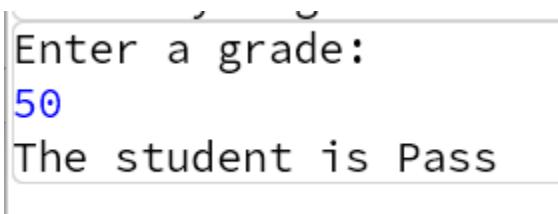
### Written and Compiled Code:



A screenshot of a Java code editor window. The menu bar at the top has buttons for Compile, Undo, Cut, Copy, Paste, Find..., and Close. The code area contains the following Java code:

```
1 import java.util.Scanner;
2 public class GradeEvaluator{
3     public static void main (String[]args){
4         Scanner gd = new Scanner (System.in);
5
6         System.out.println("Enter a grade: ");
7
8         double grade = gd.nextDouble();
9         String result = (40<=grade)? "Pass":"Fail";
10        System.out.println ("The student is "+ result);
11    }
12 }
13
14 }
```

### Actual Output



A screenshot of a terminal window. It displays the following text:

```
Enter a grade:
50
The student is Pass
```

### Question 3: Data Type Inspector

Create a Java program named `DataTypeInspector.java` that:

- Declares and initializes a variable for each of Java's 8 primitive data types.
- Uses appropriate literal values for initialization.
- Prints the value of each variable to the console, each with a descriptive

Label.

### Test case

#### Test case

Objective	To compile and execute <code>DataInspectorType.java</code> and inspecting all the different datatypes used in java
Action	<ul style="list-style-type: none"><li>• Bluej was opened</li><li>• <code>DataInspectorType.java</code> class was formed</li><li>• Inspected all the datatype</li><li>• Compiled the program with necessary comments alongside for readability.</li><li>• Successfully executed the program.</li></ul>
ExpectedResult	No compile-time errors and displayed the output.
ActualResult	Program ran smoothly as expected without any errors
Conclusion	Successful test results obtained

## Written and Compiled Code

```
1 public class DataTypeInspector{  
2     public static void main (String [] args){  
3  
4         boolean b=true;  
5         byte a= 10;  
6         short c= 50;  
7         int i= 60;  
8         long l= 12L;  
9         float f= 30.3f;  
10        double d= 22.5d;  
11        char ch= 'A';  
12  
13        System.out.println("The value of boolean is " + b);  
14        System.out.println("The value of byte is " + 10);  
15        System.out.println("The value of short is " + 50);  
16        System.out.println("The value of int is " + i);  
17        System.out.println("The value of long is " + l);  
18        System.out.println("The value of float is " + f);  
19        System.out.println("The value of double is " + d);  
20        System.out.println("The value of character is " +ch);  
21  
22  
23    }  
24}  
25}
```

## Actual Output

```
The value of boolean is true  
The value of byte is 10  
The value of short is 50  
The value of int is 60  
The value of long is 12  
The value of float is 30.3  
The value of double is 22.5  
The value of character is A
```

## Question 4: Default Value Checker

Create a Java class named DefaultValues.java

- Declare member variables (fields) for all 8 primitive types without initializing them.
- In the main method, create an instance of the class and print the value of each field.
- Add a comment explaining why this wouldn't work for local variables.

### Test case

#### Test case

Objective	To compile and execute DefaultValueChecker.java and checking out the default values of each datatype in java.
Action	<ul style="list-style-type: none"><li>• Bluej was opened</li><li>• DefaultValues.java class was formed</li><li>• Inspected the default value of all datatypes</li><li>• Compiled the program with necessary comments alongside for readability.</li><li>• Successfully executed the program.</li></ul>
ExpectedResult	No compile-time errors and displayed the output.
ActualResult	Program ran smoothly as expected without any errors
Conclusion	Successful test results obtained

## Written and Compiled Code:

```
1 public class DefaultValues{  
2     boolean b;  
3     byte a;  
4     short c;  
5     int i;  
6     long l;  
7     float f;  
8     double d;  
9     char ch;  
10      
11    public static void main (String [] args){  
12        DefaultValues obj = new DefaultValues(); //creating a obj  
13        System.out.println ("value of int "+ obj.i);  
14        System.out.println ("Default value of all data types:\n "+ obj.i + obj.a + obj.c + obj.l + obj.ch + obj.b + obj.d +obj.f);  
15          
16        // This wont work for local variables because local variables  
17        //need a literal to work or else it will give compile time error  
18    }  
19}  
20}
```

## Actual Output:

```
value of int 0  
Default value of all data types:  
0000false0.00.0
```

## Question 5: Literal Practice

Create a program named LiteralPractice.java that demonstrates the use of specific literals:

- A long variable initialized with a value requiring the 'L' suffix.
- A float variable initialized with a value requiring the 'f' suffix.
- A char variable initialized using a Unicode escape sequence (e.g., for the copyright symbol ©).
- Print the value of each variable.

### Test case

#### Test case

Objective	To compile and execute LiteralPractice.java and practicing the different literal types like char(unicode)
Action	<ul style="list-style-type: none"><li>• Bluej was opened</li><li>• LiteralPractice.java class was formed</li><li>• Practiced 3 literal types (Float, Double &amp; Char)</li><li>• Compiled the program with necessary comments alongside for readability.</li><li>• Successfully executed the program.</li></ul>
ExpectedResult	No compile-time errors and displayed the output.
ActualResult	Program ran smoothly as expected without any errors
Conclusion	Successful test results obtained

## Written and Compiled Code:

```
1 public class LiteralPractice
2 {
3     public static void main (String[]args){
4         long l= 25L;
5         float f = 25.2f;
6         char unicode = '\u00a9';
7         System.out.println ("Long variable: " +l);
8         System.out.println ("float variable: " +f);
9         System.out.println ("Char Unicode:" + unicode);
10    |
11 }
12 }
13 }
14 }
```

## Actual Code:

```
Long variable: 25
float variable: 25.2
char Unicode:@
```

## #5 Scenario Question

### Context

A local rickshaw service in Biratnagar needs a simple tool to calculate fares for their customers. The fare calculation has a few components: a base fare, a per-kilometer charge, and a per-minute charge. They also offer discounts for locals on long distances and have a surcharge for night-time travel.

### Problem

The rickshaw drivers need a program that can:

- Take distance (in km) and time (in minutes) as input.
- Ask if the customer is a local and if the travel is during the night. (Hint: use ternary operator)
- Calculate the total fare based on the rules.
- Display the final fare in a clear, Nepali format (e.g., "Rs. 550").

### Solution

The RickshawFare.java program will be a console-based Java application. It will prompt the user for the necessary inputs and then calculate and display the total fare. This will ensure consistent and transparent pricing for all customers.

### Execution

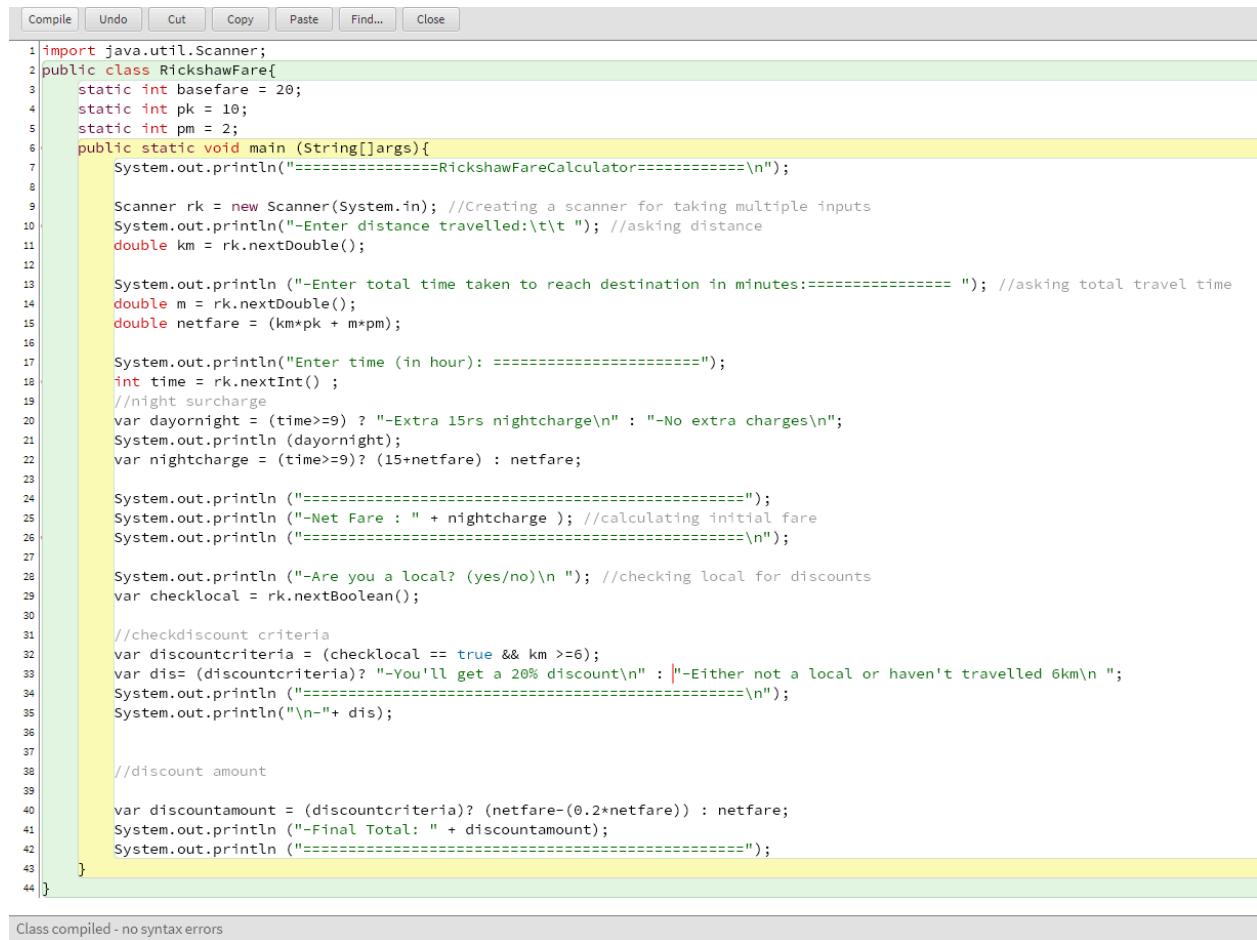
- The program is executed from the command line (e.g., Java RickshawFare).
- The program will interactively ask for inputs.
- The output will be the final calculated fare, displayed in the console.

## Test case

### Test case

Objective	To compile and execute RickshawFare.java and format it to make it easier to understand for the customer and the ridegiver.
Action	<ul style="list-style-type: none"> <li>• Bluej was opened</li> <li>• RickshawFare.java class was formed</li> <li>• Created an algorithm for the code and wrote the code with proper comments and notations to make it user-friendly.</li> <li>• Successfully executed the program and tested it with every possible situation to make it bug free.</li> </ul>
ExpectedResult	No compile-time errors and displayed the output.
ActualResult	Program ran smoothly as expected without any errors
Conclusion	Successful test results obtained

## Written and Compiled Code:



```

1 import java.util.Scanner;
2 public class RickshawFare{
3     static int basefare = 20;
4     static int pk = 10;
5     static int pm = 2;
6     public static void main (String[]args){
7         System.out.println("=====RickshawFareCalculator=====\n");
8
9         Scanner rk = new Scanner(System.in); //Creating a scanner for taking multiple inputs
10        System.out.print("Enter distance travelled:\t"); //asking distance
11        double km = rk.nextDouble();
12
13        System.out.print("Enter total time taken to reach destination in minutes:==== "); //asking total travel time
14        double m = rk.nextDouble();
15        double netfare = (km*pk + m*pm);
16
17        System.out.print("Enter time (in hour): =====");
18        int time = rk.nextInt();
19        //night surcharge
20        var dayornight = (time>=9) ? "-Extra 15rs nightcharge\n" : "-No extra charges\n";
21        System.out.println(dayornight);
22        var nightcharge = (time>=9)? (15+netfare) : netfare;
23
24        System.out.println ("=====");
25        System.out.println ("-Net Fare : " + nightcharge ); //calculating initial fare
26        System.out.println ("=====\\n");
27
28        System.out.print("Are you a local? (yes/no)\n ");
29        var checklocal = rk.nextBoolean();
30
31        //checkdiscount criteria
32        var discountcriteria = (checklocal == true && km >=6);
33        var dis= (discountcriteria)? "-You'll get a 20% discount\n" : "-Either not a local or haven't travelled 6km\n";
34        System.out.println ("=====\\n");
35        System.out.println("\n"+ dis);
36
37
38        //discount amount
39
40        var discountamount = (discountcriteria)? (netfare-(0.2*netfare)) : netfare;
41        System.out.println ("Final Total: " + discountamount);
42        System.out.println ("=====");
43    }
44 }

```

Class compiled - no syntax errors

## Actual Output:

When not a local:

```
=====RickshawFareCalculator=====

-Enter distance travelled:
35
-Enter total time taken to reach destination in minutes:
45
Enter time (in hour):
10.00
-Extra 15rs nightcharge

=====
-Net Fare : 455.0
=====

-Are you a local? (true/false)

false
=====

--Either not a local or haven't travelled 6km

-Final Total: 440.0
=====
```

When a local:

```
=====RickshawFareCalculator=====

-Enter distance travelled:
40
-Enter total time taken to reach destination in minutes:
60
Enter time (in hour):
10
-Extra 15rs nightcharge

=====
-Net Fare : 535.0
=====

-Are you a local? (true/false)

true
=====

--You'll get a 20% discount

-Final Total: 416.0
=====
```