CSN-352

CP1 PROJECT REPORT

CONTENTS OF THE REPORT:

1. TEAM MEMBERS: GROUP NO-36

Kudikala Rishikesh(20114046)
Rahul Kurkure(20114079)
Vikas Dheravath(20114033)

2. PROJECT PROPOSAL: PROJECT NO-9

Write a parser in Java language that accepts code in C++ and checks for syntax errors. (Expected: loops, integers, if-else, Boolean, char, string).

3. RESULTS:

Our code is capable to the following:

1. Checks headers of type "#include<link>" and "using namespace std;" .
2. Checks for a main function with return type void (i.e., no return type)
3. Checks parenthesis balancing (Parenthesis allowed are (,), { & })
4. Checks the variable declared is already declared in the same scope or not. Gives an error if declared in the same scope.
5. Checks Statements like "declarations" (Ex: String a = b; , char ch = 34; , etc..).
6. Checks Boolean expressions- evaluates whether the format is correct or not if they are present in while, for, if conditions.
7. Checks "Assignment" statements (Ex: a =5; fb += a+b*c; k -= u*f&c , etc..).
8. Can handle unary statements like a++, b--.
9. Also checks whether the "assignment" variables are declared previously or not.
10. Checks for the proper order of variables and operators in "assignment" statements.
11. Checks "if" statement and "else" statement if present. Checks whether Boolean expression is present or not in the if condition.
12. Nested "if else" is also checked along with the statements that are present inside the if and else blocks.
13. Checks "while loops" and its format. Checks whether Boolean expression is present or not in the while condition.
14. Checks "for loops" and its format briefly. Checks the following inside for loop conditions:
    a. Whether the 1st statement is a declaration, assignment or null. Also checks for right assignment and declaration.
    b. Whether Boolean expr is present or not in 2nd statement.
    c. Whether the 3rd statement is an assignment or null. Also checks for right assignment.
15. The above mentioned can be nested in any ways and still our code checks them.

4. HOW TO RUN THE CODE:

To run the code on a test case written in program.cpp file, just run "java parser.java".

5. WORKING OF CODE: THE FLOW

Brief explanation of all the functions in my code is given below. Here I talk about the flow of our code. The entire code is in one file “parser.java” file and within one class.

- ✓ First when the code is run, the main function starts reading the test case file – “program.cpp”.
- ✓ It takes all the document as string and breaks them into individual string lines.
- ✓ Next it removes single line comments if present.
- ✓ Removes multi-line comments.
- ✓ Now starts reading the lines one by one and checks for the header format if header is present using “checkheader()” function.
- ✓ Then checks for a main function with no return type or say return type void.
- ✓ After this main function calls a function named “checkstatements()” which is the function responsible for recursively checking all the statements in our testcase.
- ✓ We use “gettoken()” function to get the next token to analyse.
- ✓ Parenthesis balancing is done by two functions “checkop()”and “checkcp()” whenever we come across the parenthesis.
- ✓ If parentheses doesn’t match or there is an imbalance in the parenthesis, it is reported.
- ✓ Whenever we are in the “checkstatements()” function we get the next token and check if the token is something we already knew.
  - o It checks if it is “” to stop execution of the program.
  - o If it is some “parenthesis”, it handles them.
  - o If it is “for” to call “checkfor()” function to check for loop format.
  - o If it is “while” to call “checkwhile()” function to check while loop format.
  - o If it is “if” to call “checkif()” function to check if format.
  - o If it is “else” then it gives an error of else without if.
  - o If it is not among them it calls “checkdeclarations()” function and checks for an int, string, char, float, double, long, for declaration. If the function returns true, then it is a declaration else it checks for others.
  - o It checks if it is an assignment statement, calls “checkassignstmt()” function to check it.
  - o If none of them, it prints an unknown statement error and continues.
  - o In loops the Boolean condition is evaluated by “checkbooleanexpr()” function.
- ✓ After checking for loop, while loop, if statement, it internally calls “checkstatements()” function to evaluate the inner statements which makes nested things possible.
- ✓ This way recursively all the statements are evaluated.
- ✓ Note: “clearscopevar()” function is used to clear the variables which have been declared internally inside for, while, etc.

6. BREIF EXPLANATION OF ALL FUNCTIONS:

- main() : starts reading the test case file – "program.cpp". It takes all the document as string and breaks them into individual string lines. Next it removes single line comments if present. Removes multi-line comments. Now starts reading the lines one by one and checks for the header format if header is present. Then checks for a main function with no return type or say return type void. After this main function calls a function named "checkstatements()" which is the function responsible for recursively checking all the statements in our testcase.

- checkstatements() : we get the next token and check if the token is something we already knew.
    - It checks if it is "" to stop execution of the program.
    - If it is some "parenthesis", it handles them.
    - If it is "for" to call "checkfor()" function to check for loop format.
    - If it is "while" to call "checkwhile()" function to check while loop format.
    - If it is "if" to call "checkif()" function to check if format.
    - If it is "else" then it gives an error of else without if.
    - If it is not among them it calls "checkdeclarations()" function and checks for an int, string, char, float, double, long, for declaration. If the function returns true, then it is a declaration else it checks for others.
    - It checks if it is an assignment statement, calls "checkassignstmt()" function to check it.
    - If none of them, it prints an unknown statement error and continues.

- gettoken() : function to get the next token to analyse. It knows where the previous token was collected last time. Starts from there to get the next token.
    - If a character like 'ch' is not valid it prints the error.

- skipwhitespace() : Skips all the whitespace between current token and next token. It helps gettoken() function to skip all the space in between.

- checkalphanum() : Checks if the given character is any of alphabet, numbers and _. Used in checkid() and isid() functions. Returns a Boolean value.

- match() : matches the word with the current stream of string. It knows where the last token was collected. So, it starts from the end of the last token.

- checkheader() : Checks the header format if header is present.

- checkop() : When (or { parenthesis is the token this function in invoked. This function pushes them onto a stack which is maintained.

- checkcp() : When) or } parenthesis is the token this function in invoked. This function checks whether top of the stack contains the current tokens corresponding opening parenthesis or not. If yes pops it, else prints error.

- checkkw() : Checks if the given string is a known keyword or not. Returns a Boolean value.

- checkid() : Similar to isid() function but checks for long int and long long int also. It is called when declaration statements are there. It stores the scope of the now declared variable in a map. Also checks that it is not a keyword. If a keyword returns false.

- isid() : Checks whether the given string contains characters belonging to only alphabets, numbers and _ only or not. Returns a Boolean value.

- isconst() : Checks whether the given string is in the format of a number or not. Checks for floating point numbers too. Returns a Boolean value.

- checkdeclarations() : Checks the format of declarations of string and char. Calls checkotherdecl() function for other types of declarations of int, etc. Won't allow constant to be declared to string types, string to be declared to char types and other integer types, char to be declared to integer types.

- checkotherdecl() : Checks the format of declarations for integer types.

- checkvalidid() : checks whether the given identifier is valid or not. Returns false if it has already been declared in the same scope or if it is a keyword.

- checkvalidconstid() : same as checkvalidid() but also checks whether it is a constant or not.

- checkfor() : checks the format of start part of the for loop and calls checkfor1() function to check the rest of the part.

- checkfor1() : checks the format of the remaining part of the for loop and evaluates the statements inside. Called by checkstatements() function when "for" keyword is encountered.

- checkwhile() : checks the format of a while loop and evaluates the statements inside. Called by checkstatements() function when “while” keyword is encountered.

- checkassignstmt() : Checks the format of an assignment statement. Checks “Assignment” statements (Ex: a =5; f += a+b*c; k -= u*f&c , etc..). Handle unary statements like a++, b--. Also checks whether the “assignment” variables are declared previously or not. Checks for the proper order of variables and operators in “assignment” statements.

- isop() : Checks whether the given string is an operator or not.

- checkassignstmt2(): Same as checkassignstmt() function, but at start it already gets a token as input whereas in the checkassignstmt() it gets a token at the start using gettoken() function.

- checkboolexpr() : Evaluates all the Boolean expression if present in if, while, for loop conditions. If expression is not present returns false.

- clearscopevar() : Used to clear the variables declared inside a scope. Scope to be checked is given as input to the function. Called when the scope of those statements like if, for, while is completed.

- checkif() : checks the format of a while loop. Called by checkstatements() function when “while” keyword is encountered. It calls checkelse() internally.

- checkelse() : checks if else is present or not after checkif() function calls it. If present checks the format and evaluates the statements inside.

- checkstatements2() : same as checkstatements() but won’t call gettoken() to get the token initially, it is given as input to the function.

7. BREIF EXPLANATION OF ALL DATASTRUCTURES USED:

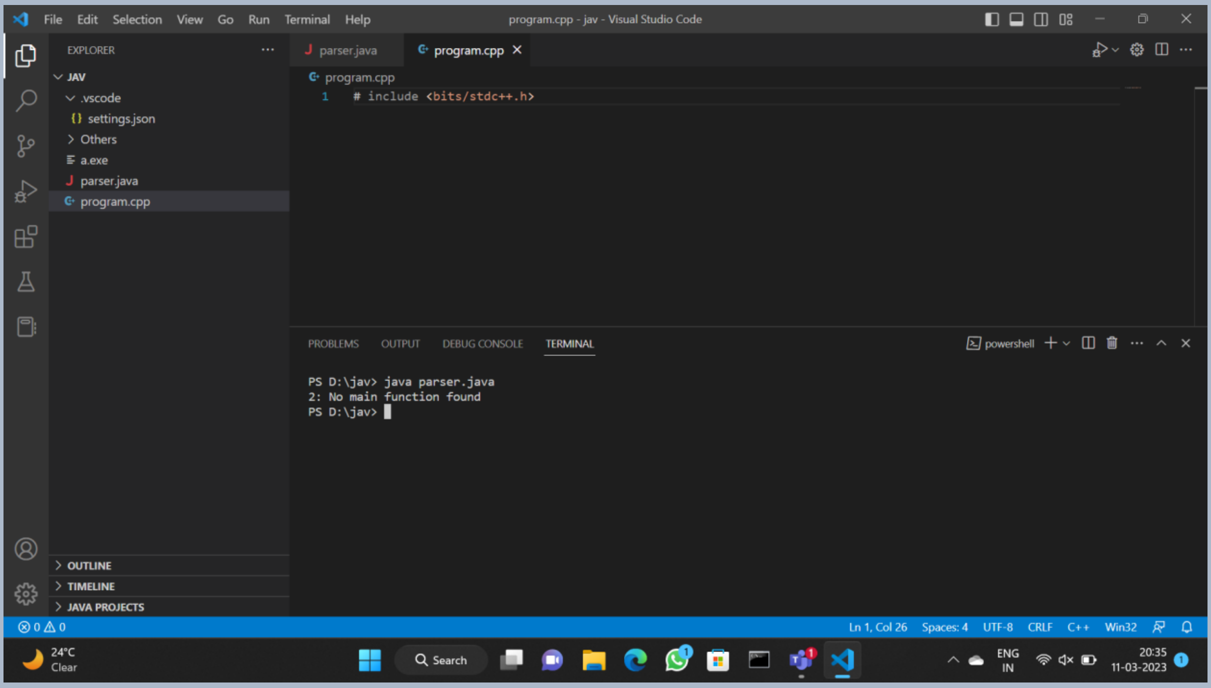
- lines[] array : to store all the string input of the testcase file “program.cpp”.

- globalmap : HashMap in java, used for storing the variables declared and their scope.

- charStack : It is used by checkop() and checkcp() to balance the parenthesis.

8. TEST CASES AND THEIR SIMULATION RESULTS:

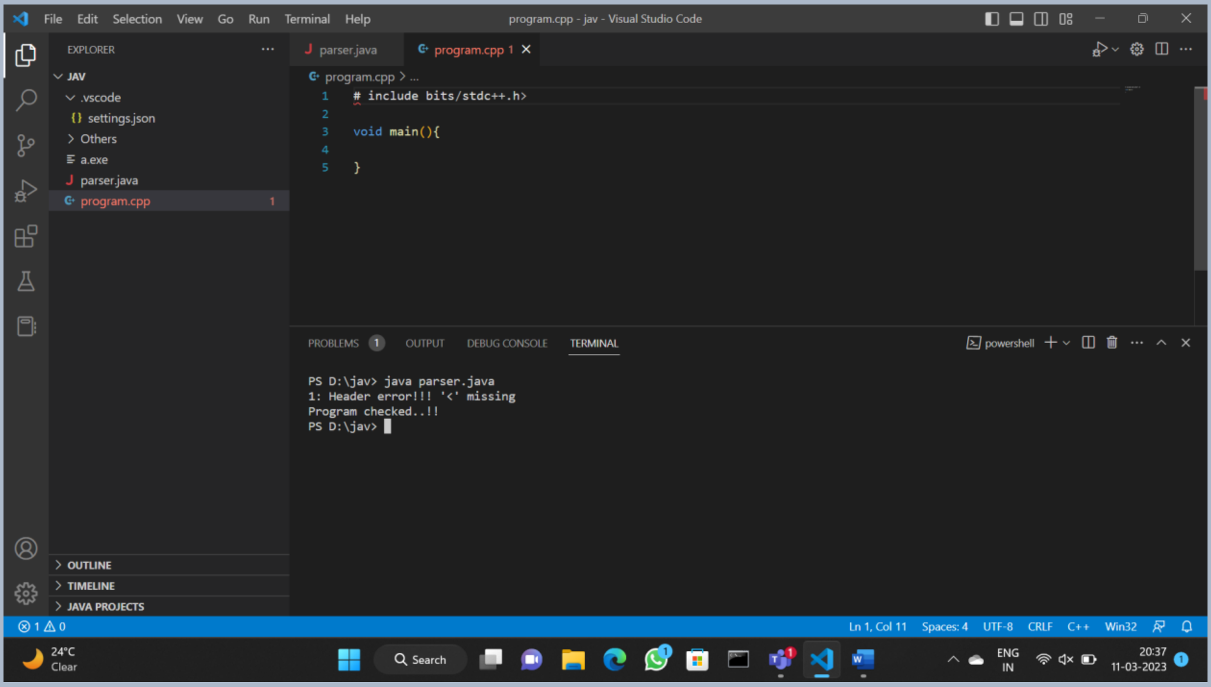
Testcase-1: Main function check.



```cpp
# include <bits/stdc++.h>
```

```
PS D:\jav> java parser.java
2: No main function found
PS D:\jav> 
```

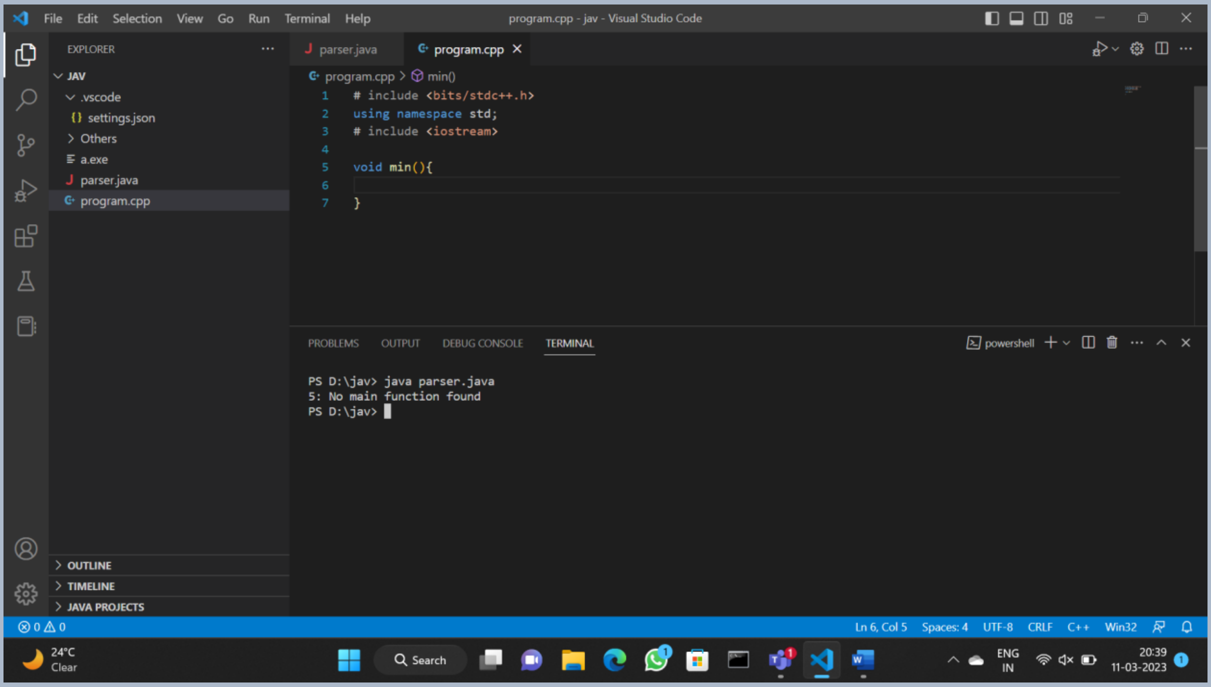
Testcase-2: Header syntax check.



```cpp
# include bits/stdc++.h>

void main(){

}
```

```
PS D:\jav> java parser.java
1: Header error!!! '<' missing
Program checked..!!
PS D:\jav> 
```

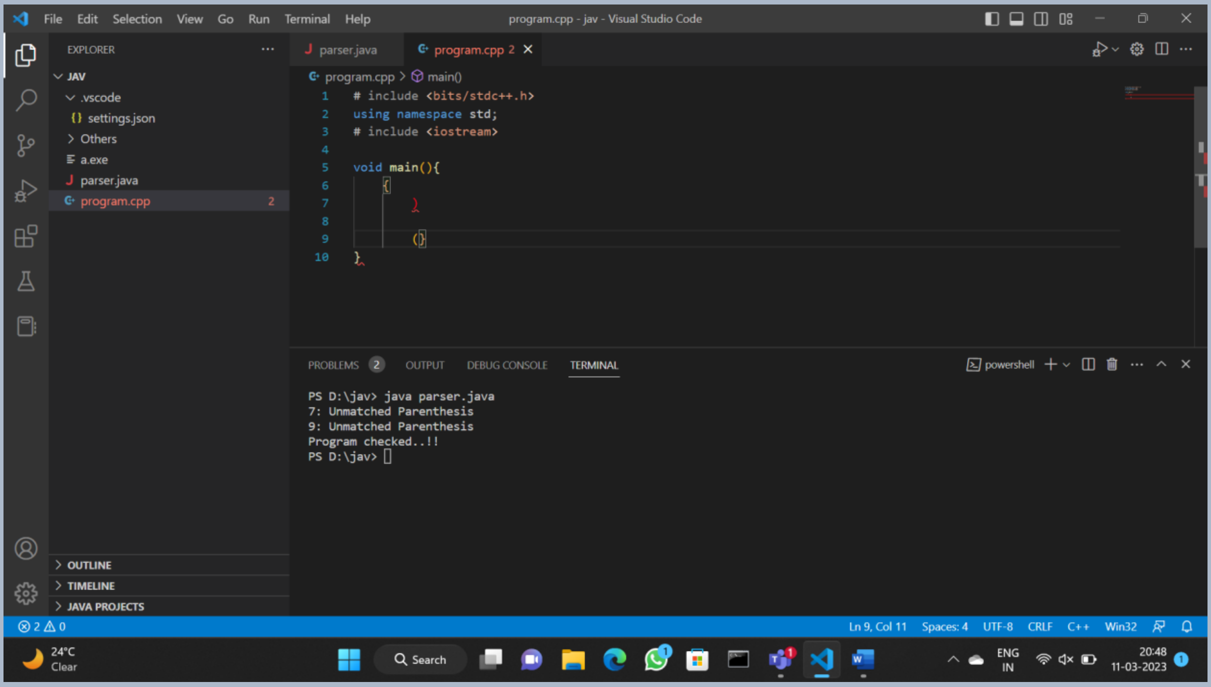
Testcase-3: Main function check.



```
# include <bits/stdc++.h>
using namespace std;
# include <iostream>

void min(){

}
```

```
PS D:\jav> java parser.java
5: No main function found
PS D:\jav> 
```

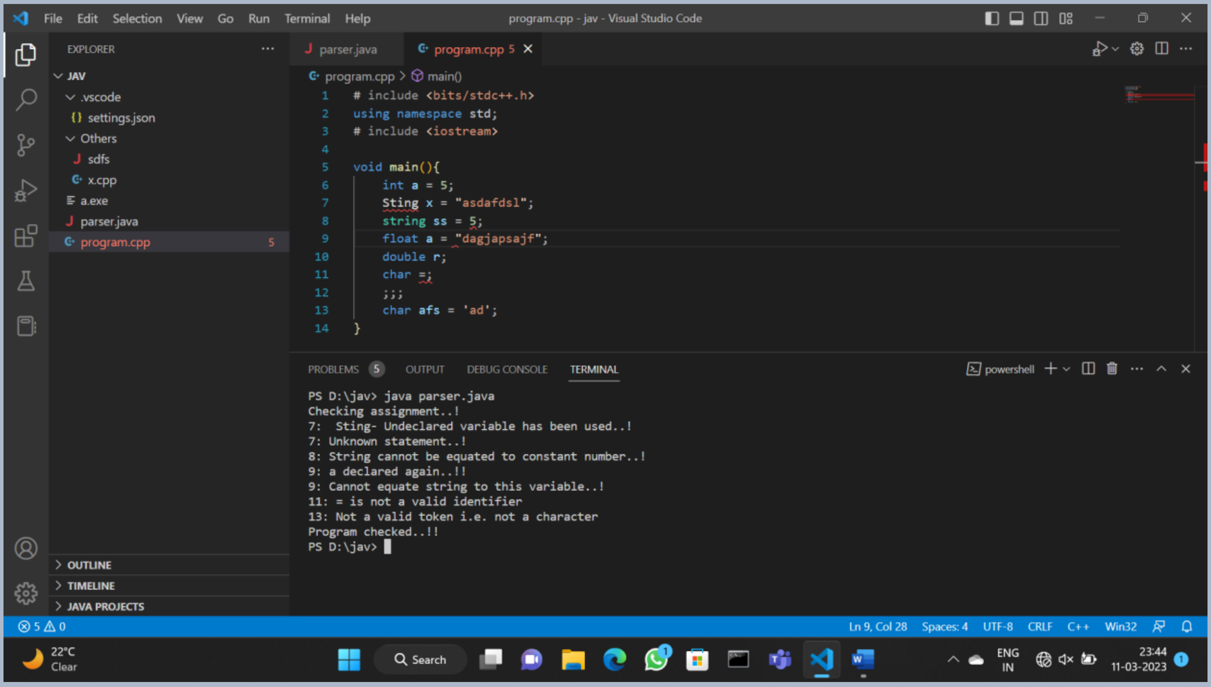
Testcase-4: Parenthesis balancing.



```
# include <bits/stdc++.h>
using namespace std;
# include <iostream>

void main(){
    {
        )

        (}
}
```

```
PS D:\jav> java parser.java
7: Unmatched Parenthesis
9: Unmatched Parenthesis
Program checked..!!
PS D:\jav> 
```

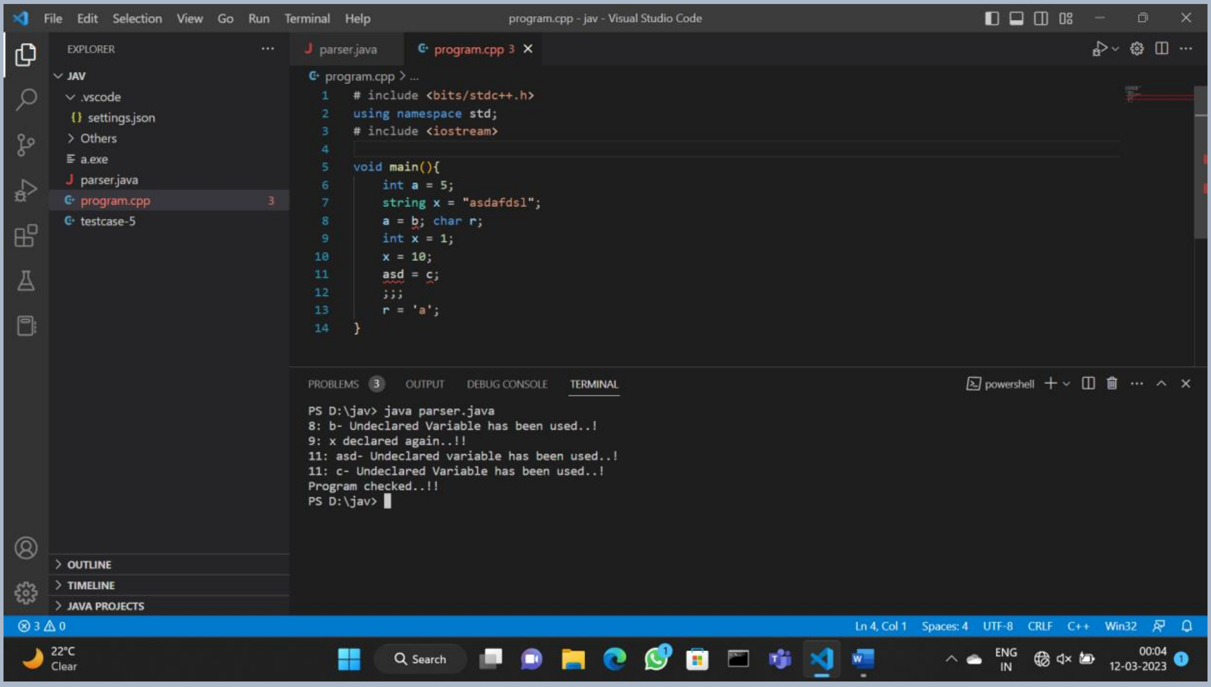
Testcase-5: Declaration statements evaluation.



```
1   # include <bits/stdc++.h>
2   using namespace std;
3   # include <iostream>
4
5   void main(){
6       int a = 5;
7       Sting x = "asdafdsl";
8       string ss = 5;
9       float a = "dagjapsajf";
10      double r;
11      char =;
12      ;;;
13      char afs = 'ad';
14  }
```

```
PS D:\jav> java parser.java
Checking assignment..!
7:  Sting- Undeclared variable has been used..!
7: Unknown statement..!
8: String cannot be equated to constant number..!
9: a declared again..!!
9: Cannot equate string to this variable..!
11: = is not a valid identifier
13: Not a valid token i.e. not a character
Program checked..!!
PS D:\jav>
```

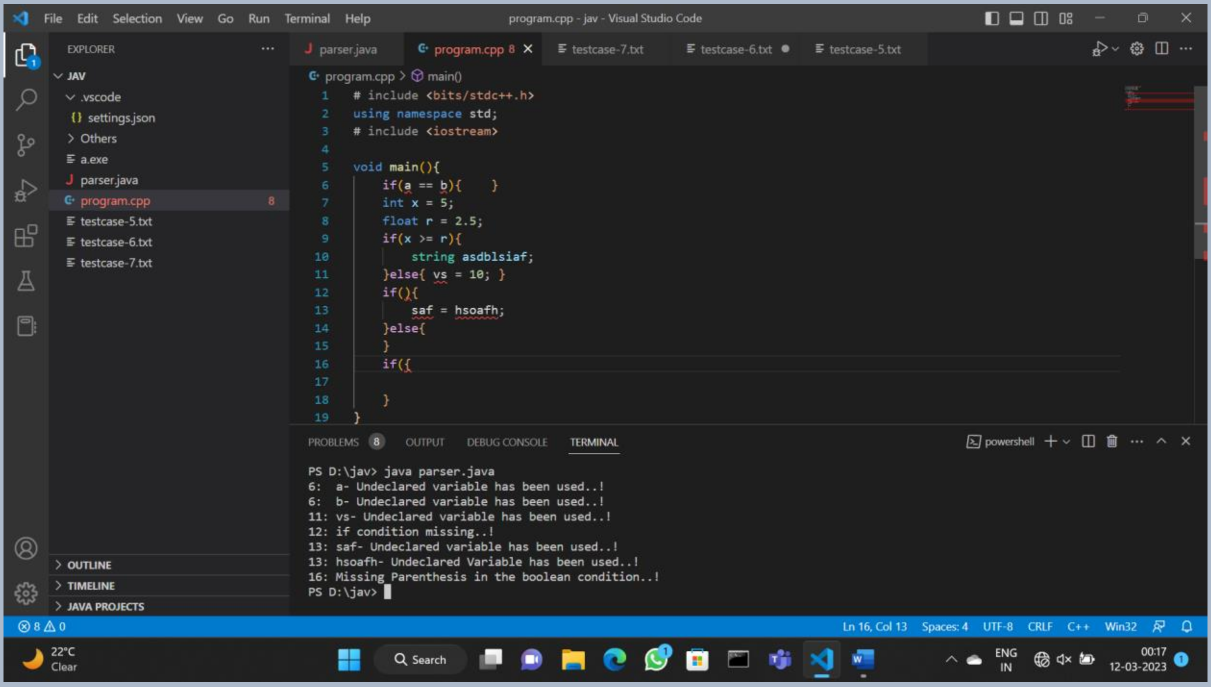
Testcase-6: Assignment and declaration statements evaluation.



```
1   # include <bits/stdc++.h>
2   using namespace std;
3   # include <iostream>
4
5   void main(){
6       int a = 5;
7       string x = "asdafdsl";
8       a = b; char r;
9       int x = 1;
10      x = 10;
11      asd = c;
12      ;;;
13      r = 'a';
14  }
```

```
PS D:\jav> java parser.java
8: b- Undeclared Variable has been used..!
9: x declared again..!!
11: asd- Undeclared variable has been used..!
11: c- Undeclared Variable has been used..!
Program checked..!!
PS D:\jav>
```

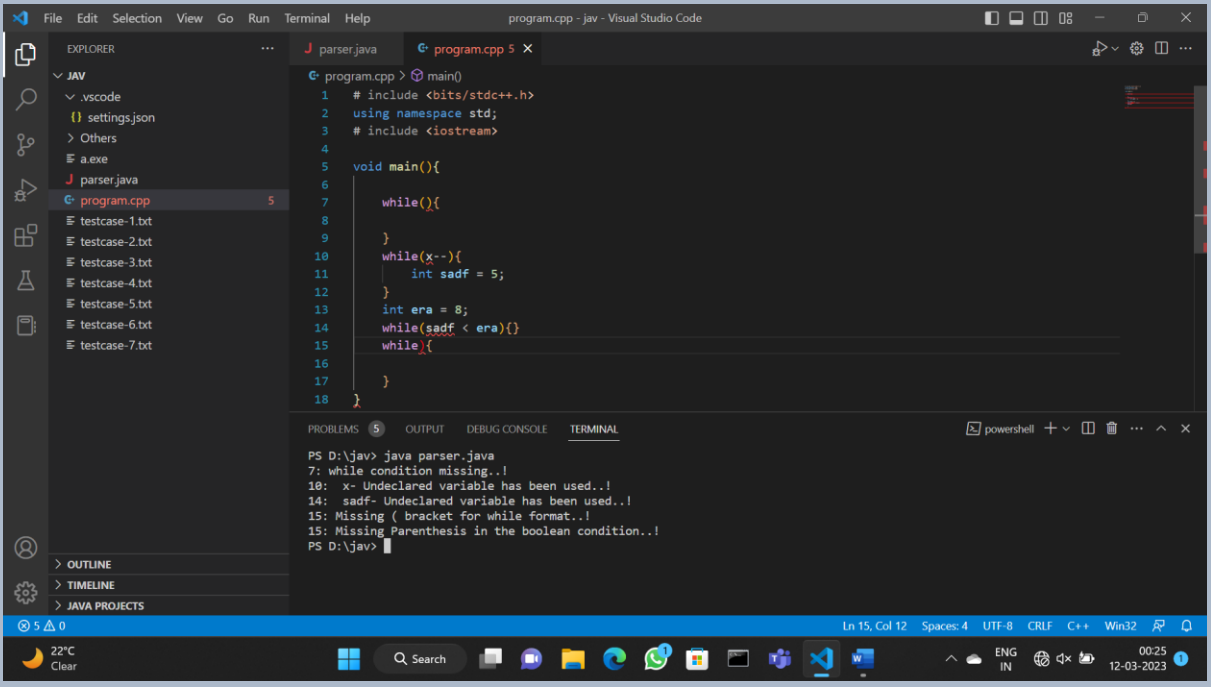
Testcase-7: If, else statements evaluation.



```
1   # include <bits/stdc++.h>
2   using namespace std;
3   # include <iostream>
4
5   void main(){
6       if(a == b){    }
7       int x = 5;
8       float r = 2.5;
9       if(x >= r){
10          string asdblsiaf;
11      }else{ vs = 10; }
12      if(){
13          saf = hsoafh;
14      }else{
15      }
16      if({
17
18      }
19  }
```

```
PS D:\jav> java parser.java
6:  a- Undeclared variable has been used..!
6:  b- Undeclared variable has been used..!
11: vs- Undeclared variable has been used..!
12: if condition missing..!
13: saf- Undeclared variable has been used..!
13: hsoafh- Undeclared Variable has been used..!
16: Missing Parenthesis in the boolean condition..!
PS D:\jav>
```

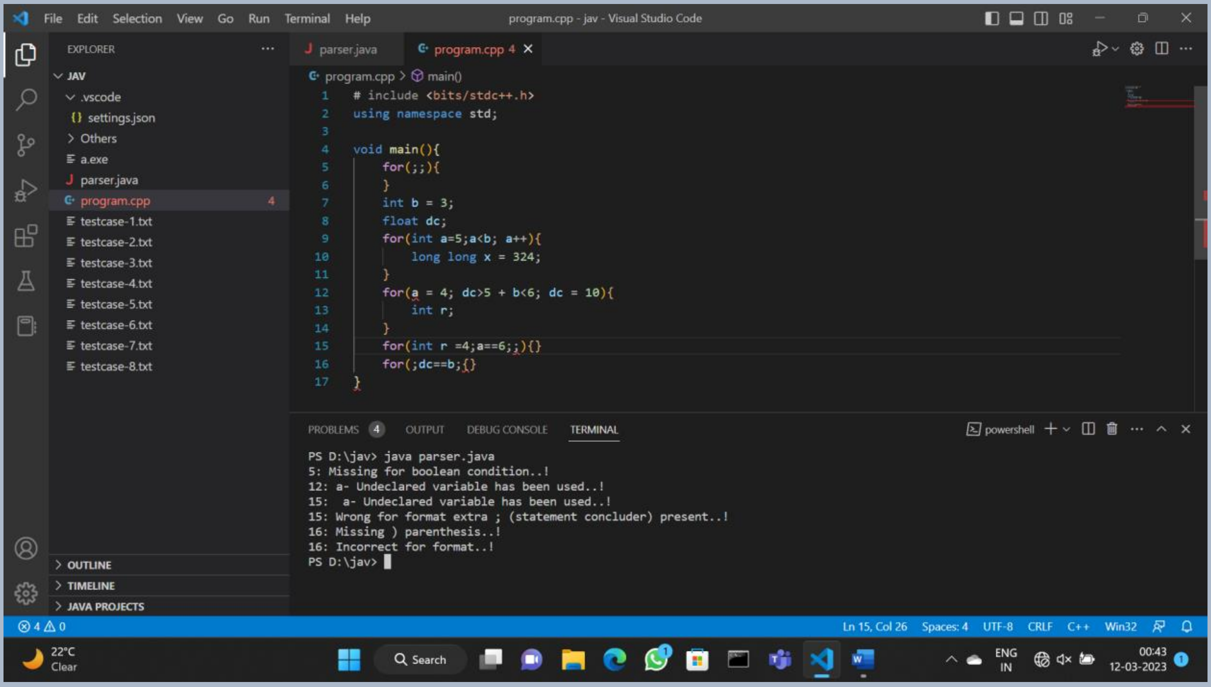
Testcase-8: While loop evaluation.



```
1   # include <bits/stdc++.h>
2   using namespace std;
3   # include <iostream>
4
5   void main(){
6
7       while(){
8
9       }
10      while(x--){
11          int sadf = 5;
12      }
13      int era = 8;
14      while(sadf < era){}
15      while){
16
17      }
18  }
```

```
PS D:\jav> java parser.java
7: while condition missing..!
10:  x- Undeclared variable has been used..!
14:  sadf- Undeclared variable has been used..!
15: Missing ( bracket for while format..!
15: Missing Parenthesis in the boolean condition..!
PS D:\jav>
```

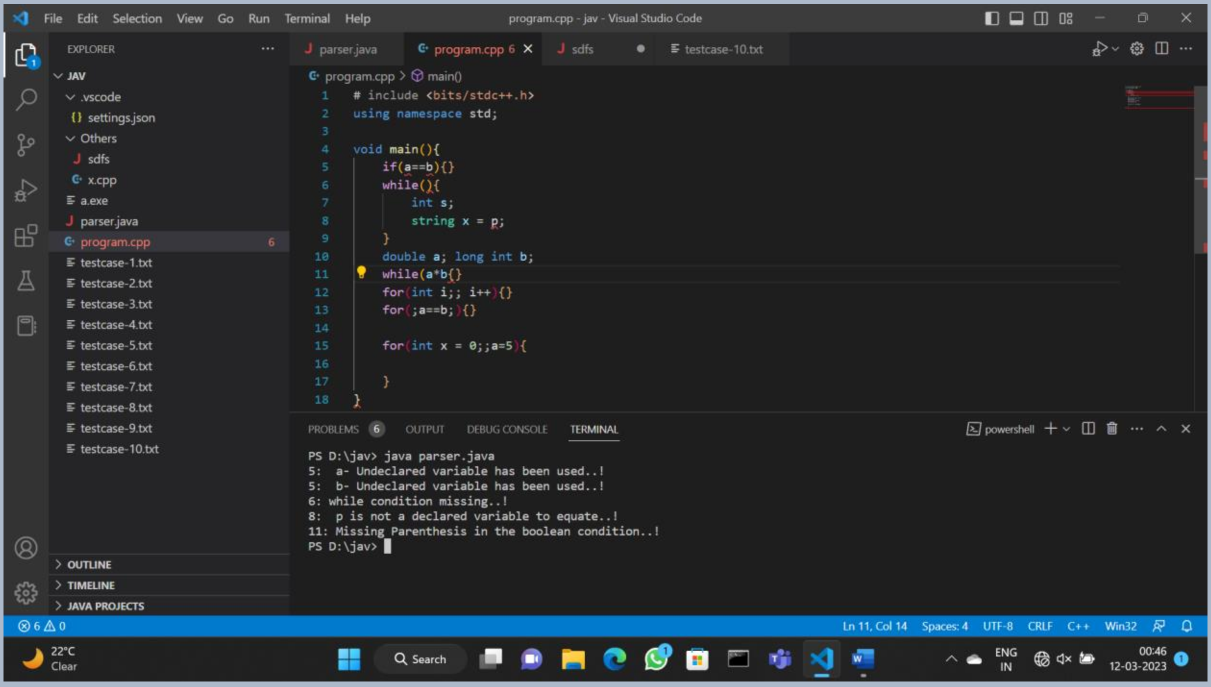
Testcase-9: For loop evaluation.



```
1   # include <bits/stdc++.h>
2   using namespace std;
3
4   void main(){
5       for(;;){
6       }
7       int b = 3;
8       float dc;
9       for(int a=5;a<b; a++){
10          long long x = 324;
11      }
12      for(a = 4; dc>5 + b<6; dc = 10){
13          int r;
14      }
15      for(int r =4;a==6;;){}
16      for(;dc==b;{}
17  }
```

```
PS D:\jav> java parser.java
5: Missing for boolean condition..!
12: a- Undeclared variable has been used..!
15:  a- Undeclared variable has been used..!
15: Wrong for format extra ; (statement concluder) present..!
16: Missing ) parenthesis..!
16: Incorrect for format..!
PS D:\jav>
```

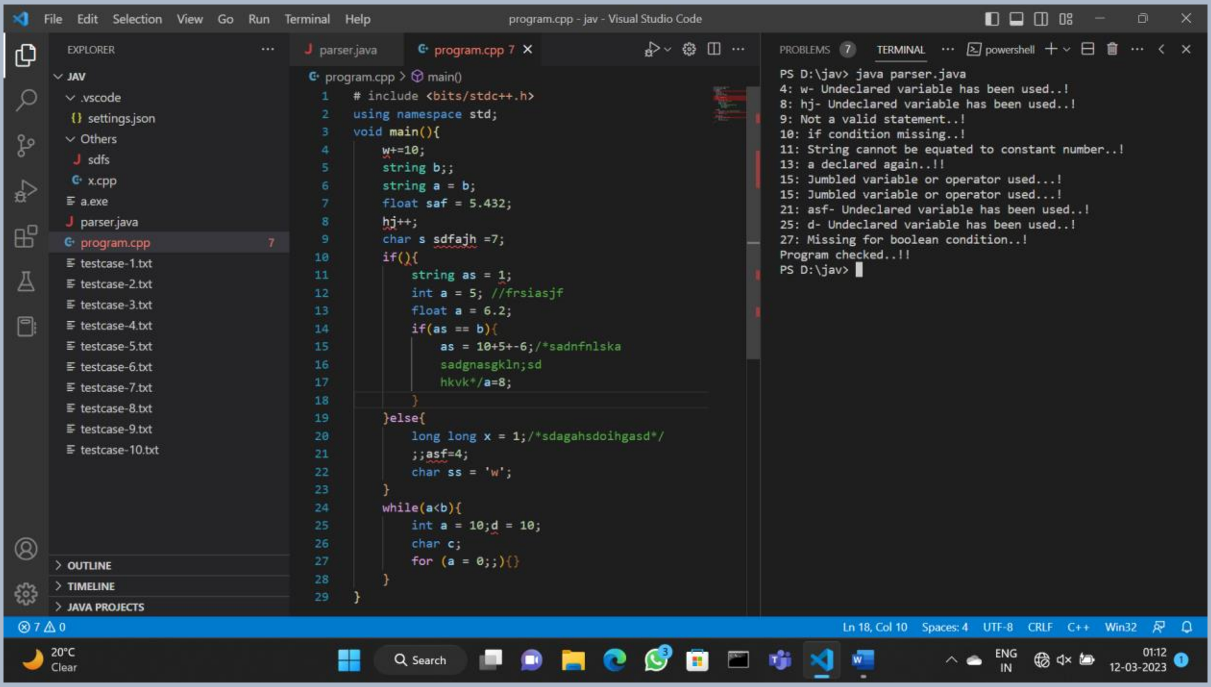
Testcase-10: For loop, while loop, if & else statements evaluation.



```
1   # include <bits/stdc++.h>
2   using namespace std;
3
4   void main(){
5       if(a==b){}
6       while(){
7           int s;
8           string x = p;
9       }
10      double a; long int b;
11      while(a*b{}
12      for(int i;; i++){}
13      for(;a==b;){}
14
15      for(int x = 0;;a=5){
16
17      }
18  }
```

```
PS D:\jav> java parser.java
5:  a- Undeclared variable has been used..!
5:  b- Undeclared variable has been used..!
6: while condition missing..!
8:  p is not a declared variable to equate..!
11: Missing Parenthesis in the boolean condition..!
PS D:\jav>
```

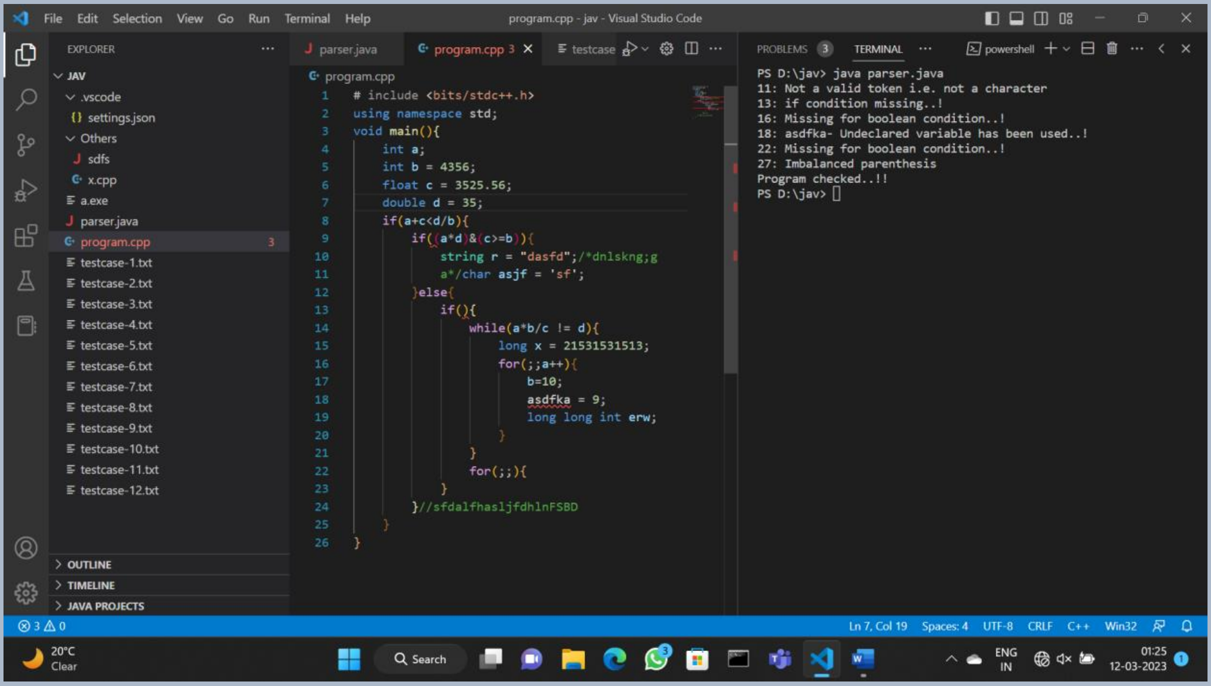
Testcase-11: Complex nested if & else, for loop , while loop, declaration, assignment statements evaluation.



```
1   # include <bits/stdc++.h>
2   using namespace std;
3   void main(){
4       w+=10;
5       string b;;
6       string a = b;
7       float saf = 5.432;
8       hj++;
9       char s sdfajh =7;
10      if(){
11          string as = 1;
12          int a = 5; //frsiasjf
13          float a = 6.2;
14          if(as == b){
15              as = 10+5+-6;/*sadnfnlska
16              sadgnasgkln;sd
17              hkvk*/a=8;
18          }
19      }else{
20          long long x = 1;/*sdagahsdoihgasd*/
21          ;;asf=4;
22          char ss = 'w';
23      }
24      while(a<b){
25          int a = 10;d = 10;
26          char c;
27          for (a = 0;;){}
28      }
29  }
```

```
PS D:\jav> java parser.java
4: w- Undeclared variable has been used..!
8: hj- Undeclared variable has been used..!
9: Not a valid statement..!
10: if condition missing..!
11: String cannot be equated to constant number..!
13: a declared again..!!
15: Jumbled variable or operator used...!
15: Jumbled variable or operator used...!
21: asf- Undeclared variable has been used..!
25: d- Undeclared variable has been used..!
27: Missing for boolean condition..!
Program checked..!!
PS D:\jav> 
```

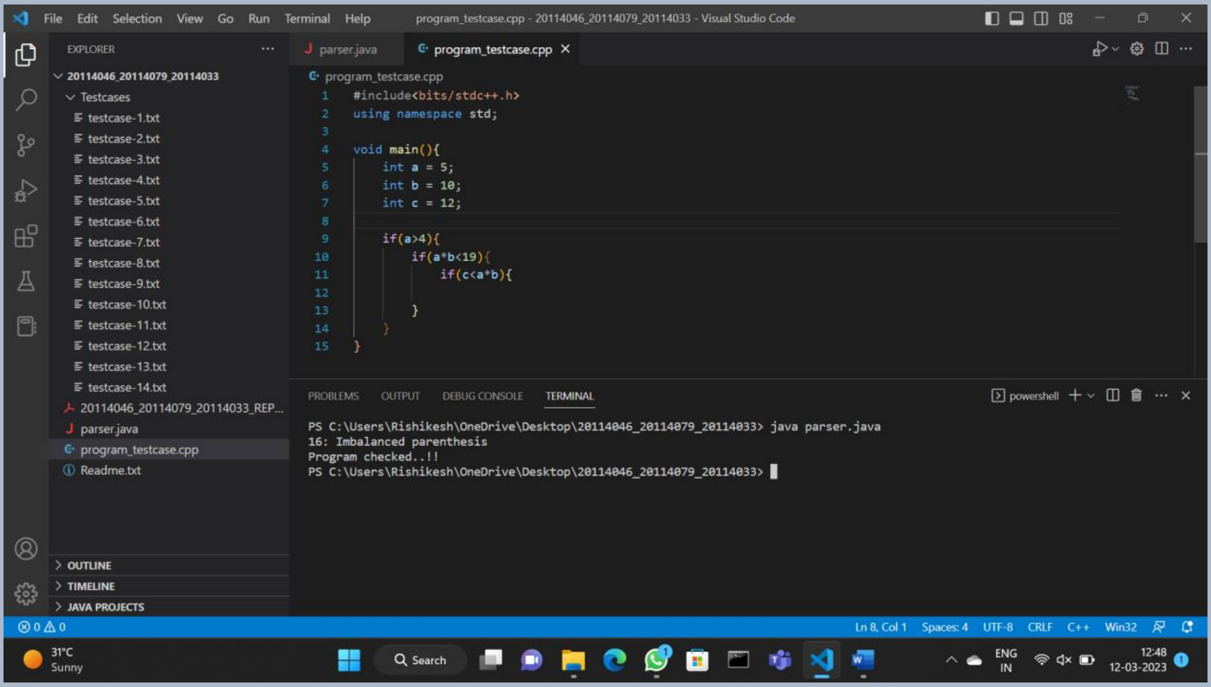
Testcase-12: Complex nested if & else, for loop , while loop, declaration, assignment statements evaluation.



```
1   # include <bits/stdc++.h>
2   using namespace std;
3   void main(){
4       int a;
5       int b = 4356;
6       float c = 3525.56;
7       double d = 35;
8       if(a+c<d/b){
9           if((a*d)&(c>=b)){
10              string r = "dasfd";/*dnlskng;g
11              a*/char asjf = 'sf';
12          }else{
13              if(){
14                  while(a*b/c != d){
15                      long x = 21531531513;
16                      for(;;a++){
17                          b=10;
18                          asdfka = 9;
19                          long long int erw;
20                      }
21                  }
22                  for(;;){
23              }
24          }//sfdalfhasljfdhlnFSBD
25      }
26  }
```

```
PS D:\jav> java parser.java
11: Not a valid token i.e. not a character
13: if condition missing..!
16: Missing for boolean condition..!
18: asdfka- Undeclared variable has been used..!
22: Missing for boolean condition..!
27: Imbalanced parenthesis
Program checked..!!
PS D:\jav> 
```

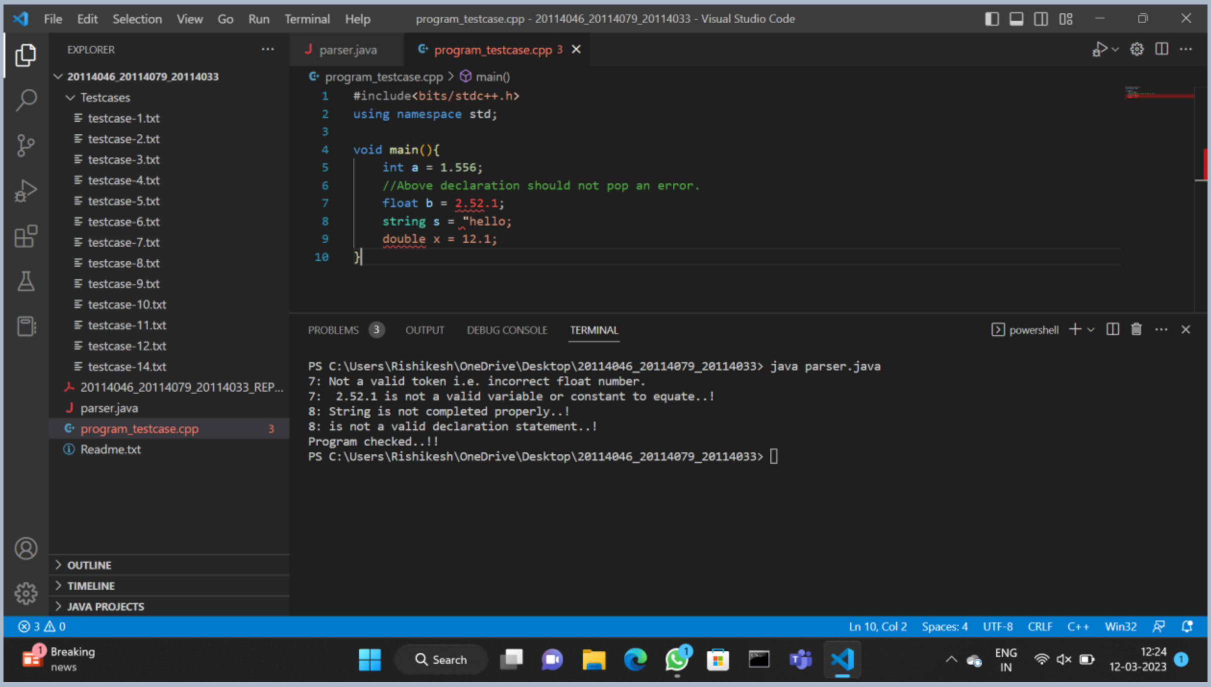
Testcase-13:



```cpp
#include<bits/stdc++.h>
using namespace std;

void main(){
    int a = 5;
    int b = 10;
    int c = 12;

    if(a>4){
        if(a*b<19){
            if(c<a*b){

        }
    }
}
```

```
PS C:\Users\Rishikesh\OneDrive\Desktop\20114046_20114079_20114033> java parser.java
16: Imbalanced parenthesis
Program checked..!!
PS C:\Users\Rishikesh\OneDrive\Desktop\20114046_20114079_20114033>
```

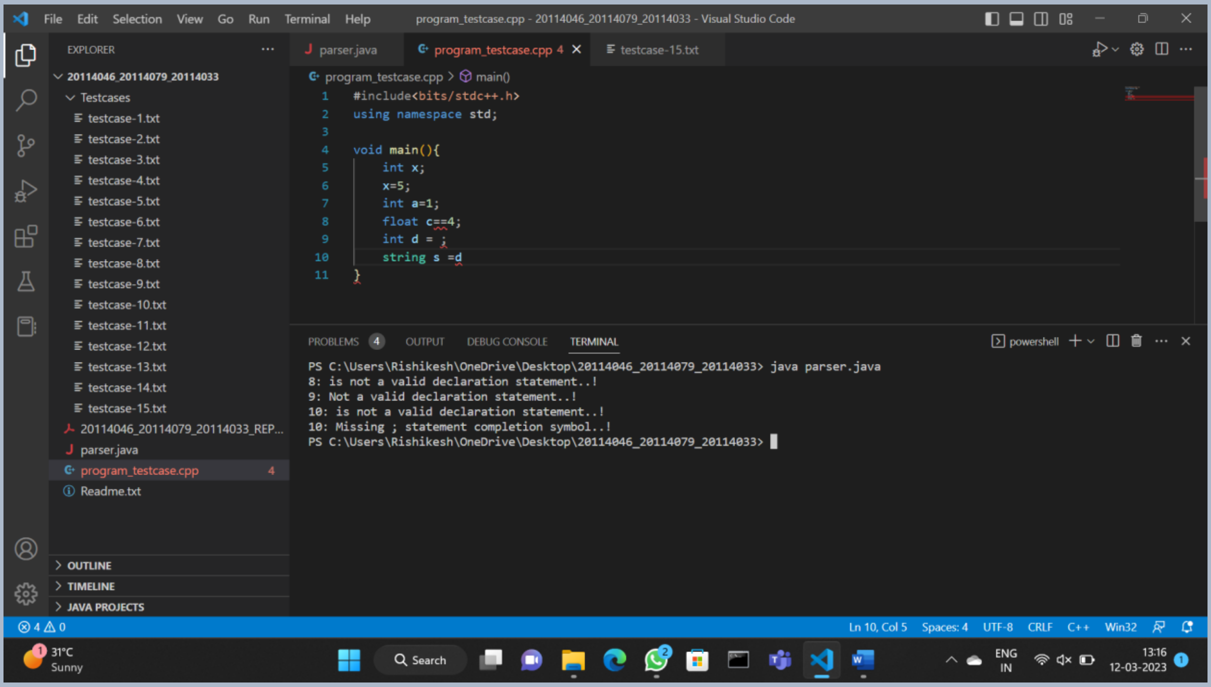
Testcase-14:



```cpp
#include<bits/stdc++.h>
using namespace std;

void main(){
    int a = 1.556;
    //Above declaration should not pop an error.
    float b = 2.52.1;
    string s = "hello;
    double x = 12.1;
}
```

```
PS C:\Users\Rishikesh\OneDrive\Desktop\20114046_20114079_20114033> java parser.java
7: Not a valid token i.e. incorrect float number.
7:  2.52.1 is not a valid variable or constant to equate..!
8: String is not completed properly..!
8: is not a valid declaration statement..!
Program checked..!!
PS C:\Users\Rishikesh\OneDrive\Desktop\20114046_20114079_20114033>
```

Testcase-15:



```cpp
1   #include<bits/stdc++.h>
2   using namespace std;
3
4   void main(){
5       int x;
6       x=5;
7       int a=1;
8       float c==4;
9       int d = ;
10      string s =d
11  }
```

```
PS C:\Users\Rishikesh\OneDrive\Desktop\20114046_20114079_20114033> java parser.java
8: is not a valid declaration statement..!
9: Not a valid declaration statement..!
10: is not a valid declaration statement..!
10: Missing ; statement completion symbol..!
PS C:\Users\Rishikesh\OneDrive\Desktop\20114046_20114079_20114033>
```

<<<<<<<<<THE END>>>>>>>>