REPORT ON JAVA INTERFACES

(COMPILER DESIGN)

Definition: An interface is a reference type in Java. It is similar to class. It is a collection of abstract methods.

Abstract method: A method without body (no implementation) is known as abstract method.

Ex : public int myMethod(int n1, int n2);

Features:

1. Along with abstract methods, an interface may also contain constants, default methods, static methods, and nested types.

2. Method bodies exist only for default methods and static methods.

3. An interface can contain any number of methods.

4. An interface is written in a file with a ".java" extension, with the name of the interface matching the name of the file.

5. The byte code of an interface appears in a ".class" file.

6. Interfaces appear in packages, and their corresponding bytecode file must be in a directory structure that matches the package name.

7. We cannot instantiate an interface.

8. An interface does not contain any constructors.

Declaring Interfaces:

The interface keyword is used to declare an interface. Here is a simple example to declare an interface −

```
* File name : NameOfInterface.java */

import java.lang.*;

// Any number of import statements

public interface NameOfInterface {

    // Any number of final, static fields
```

```
    // Any number of abstract method declarations\
}
```

Properties of Interface:

1. An interface is implicitly abstract. We do not need to use the abstract keyword while declaring an interface.

2. Each method in an interface is also implicitly abstract, so the abstract keyword is not needed.

3. Methods in an interface are implicitly public.

Java Interface Implementation

Syntax:

```
public interface NameOfInterface {
  // Any number of final, static fields
  // Any number of abstract method declarations\
}
```

Example

```
// Filename: Sports.java
public interface Sports {
  public void setHomeTeam(String name);
  public void setVisitingTeam(String name);
}
```

Things to check:

1. opening and closing braces to be balanced

2. syntax appropiately written

Limitations:

1. You can only use char, String, int and float values to declare any function

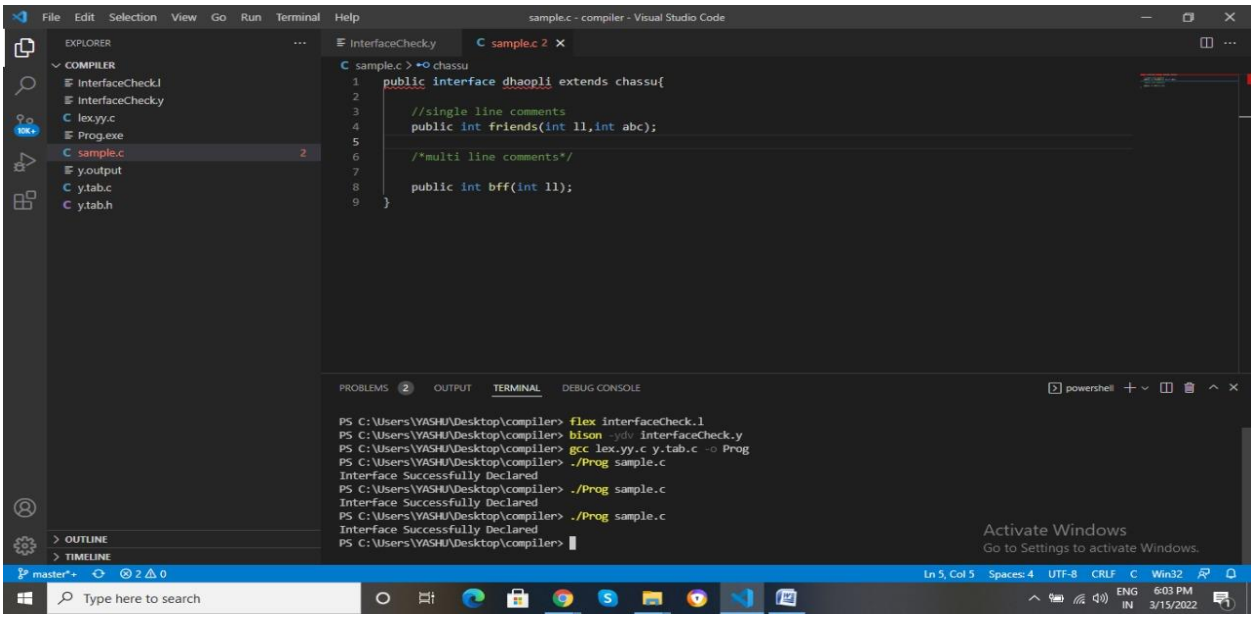
Erros not handled:

1. Anthing after closing bracket of interface is not handled.

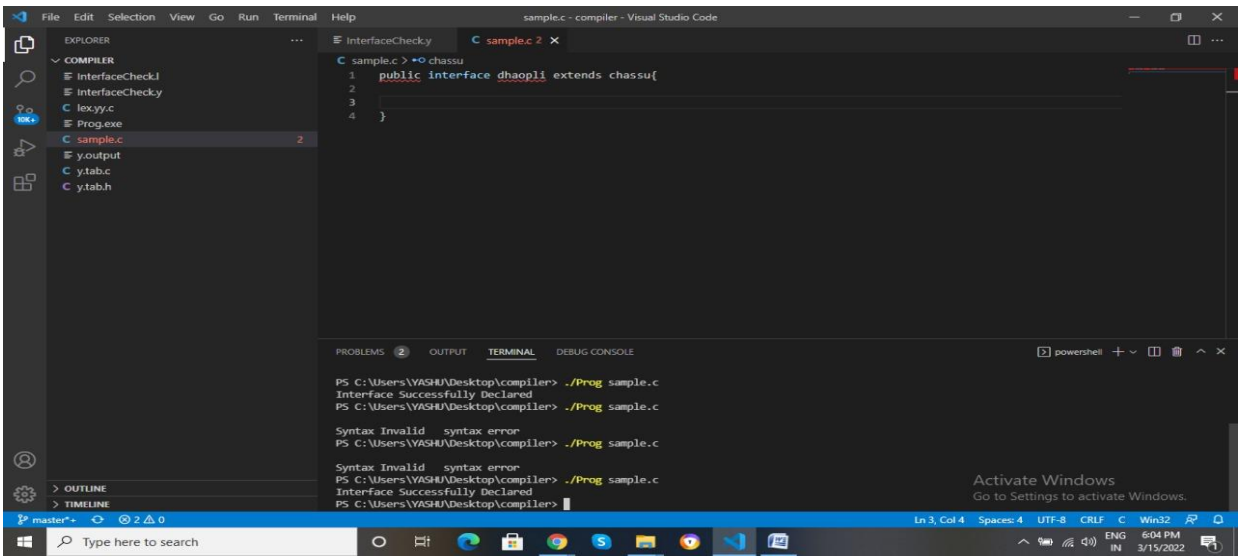
Test Cases :

Accepted test cases

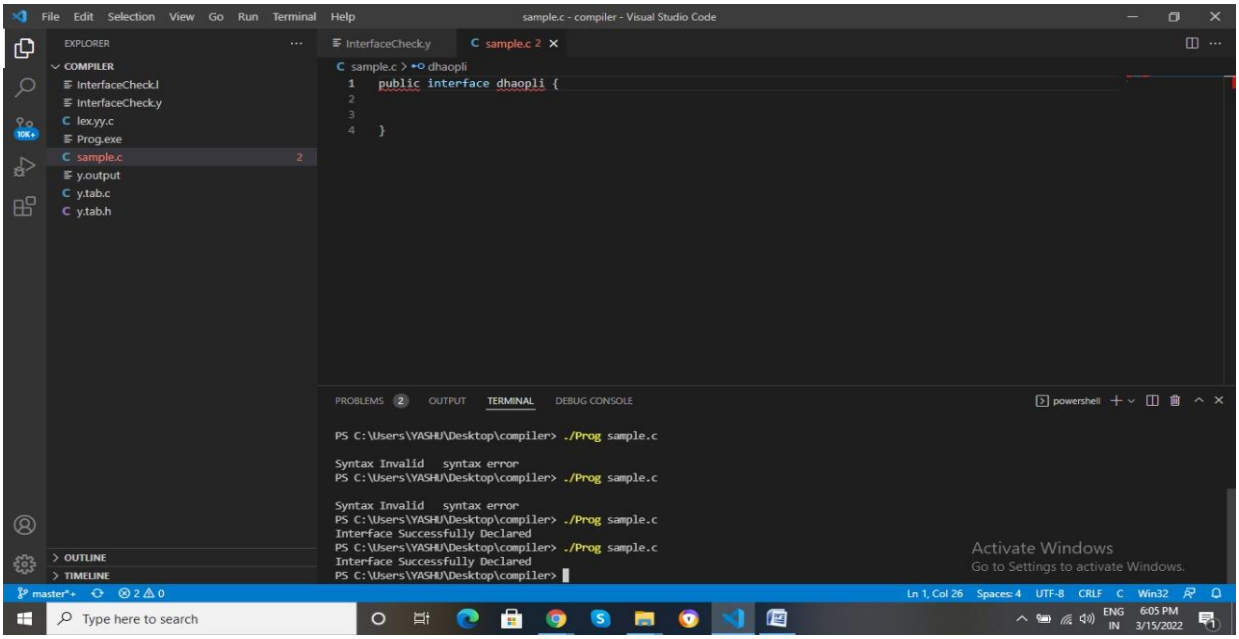
1. Handled single line and multiline comments.



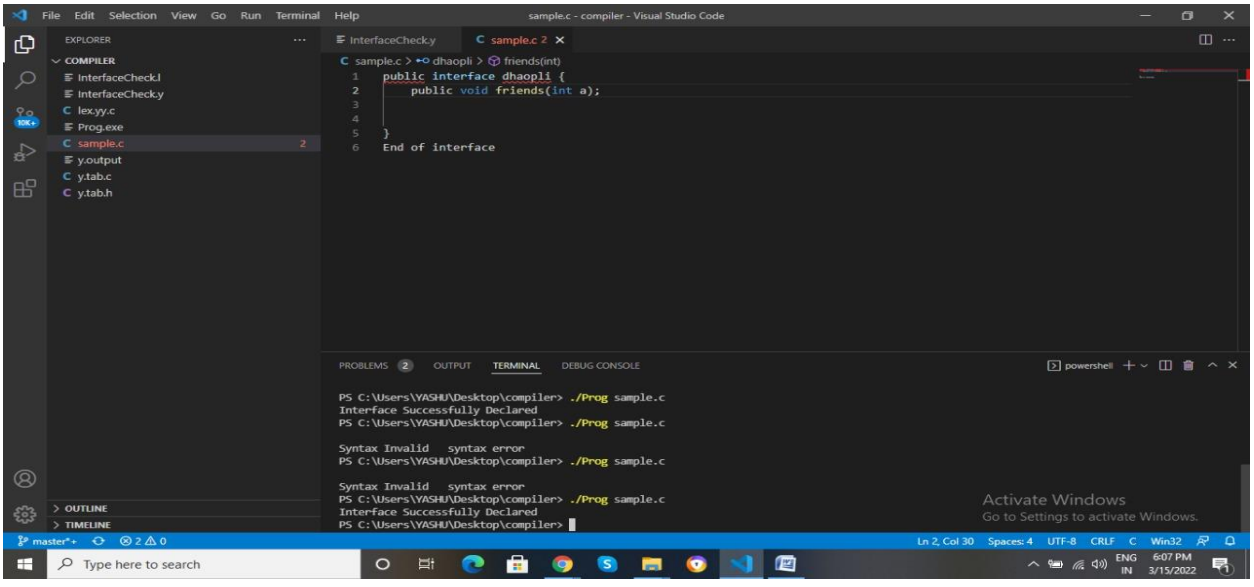
2. Handled after removing body of interface .



3. Handled without extends keyword

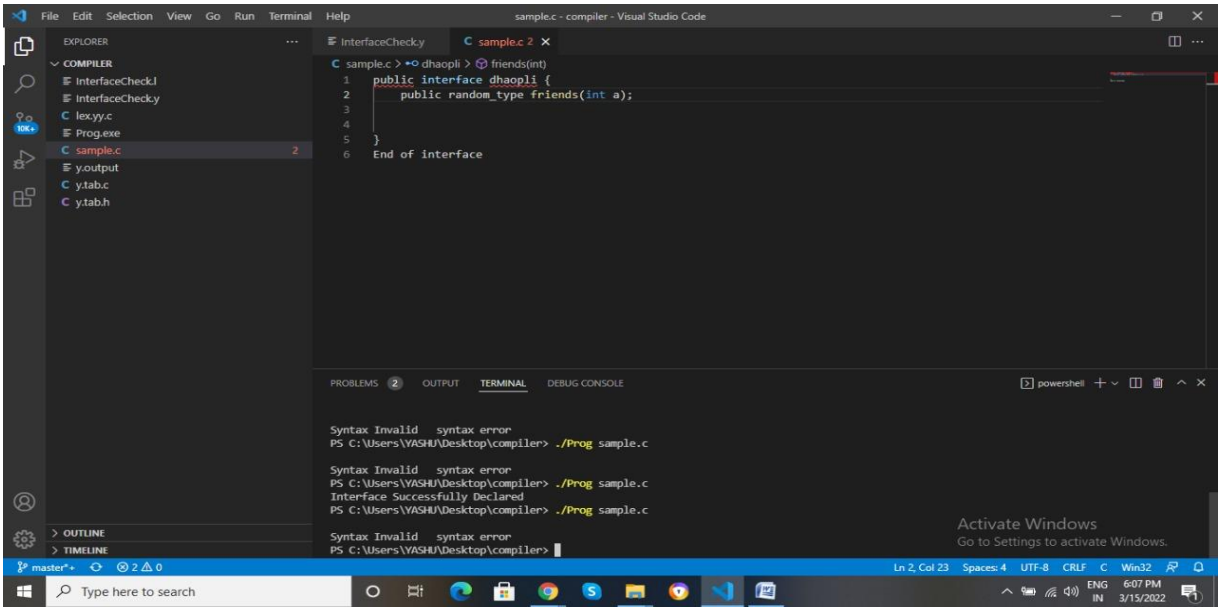


4. Not handled – anything after closing bracket of interface is accepted
   Assumptions – after declaring interface we can declare class which implements this interface , so its accepted.

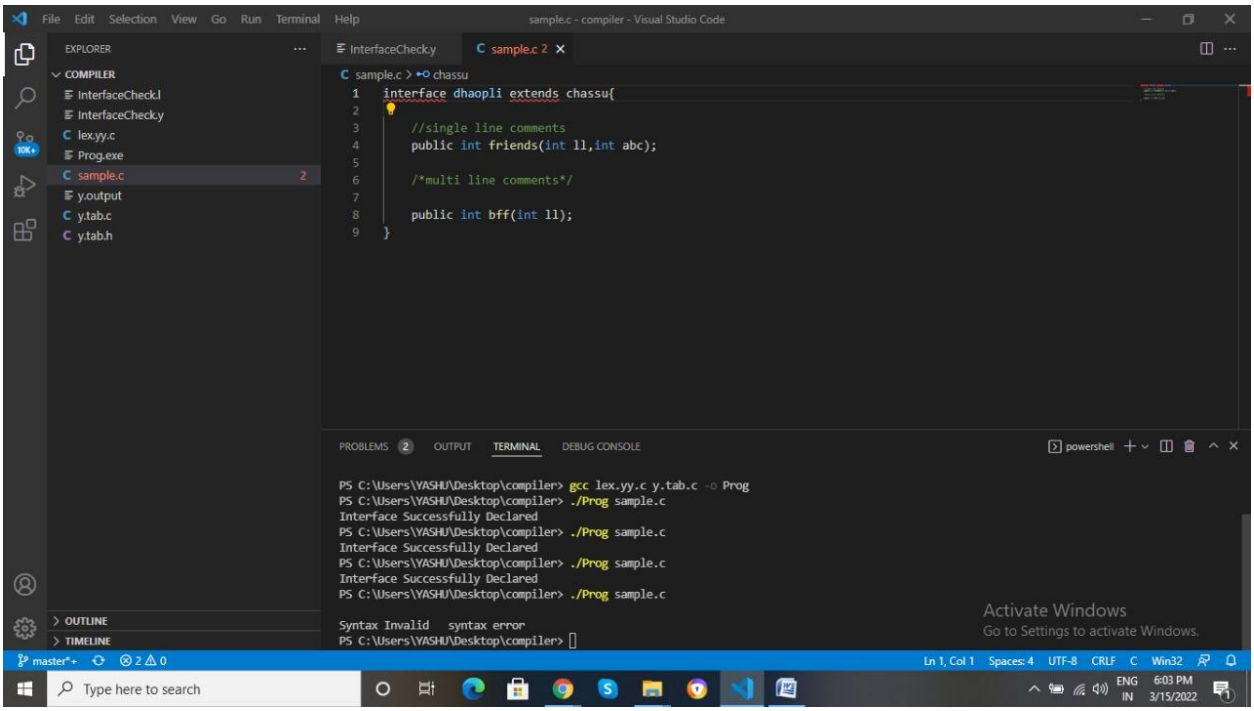


Not Accepted Test Cases

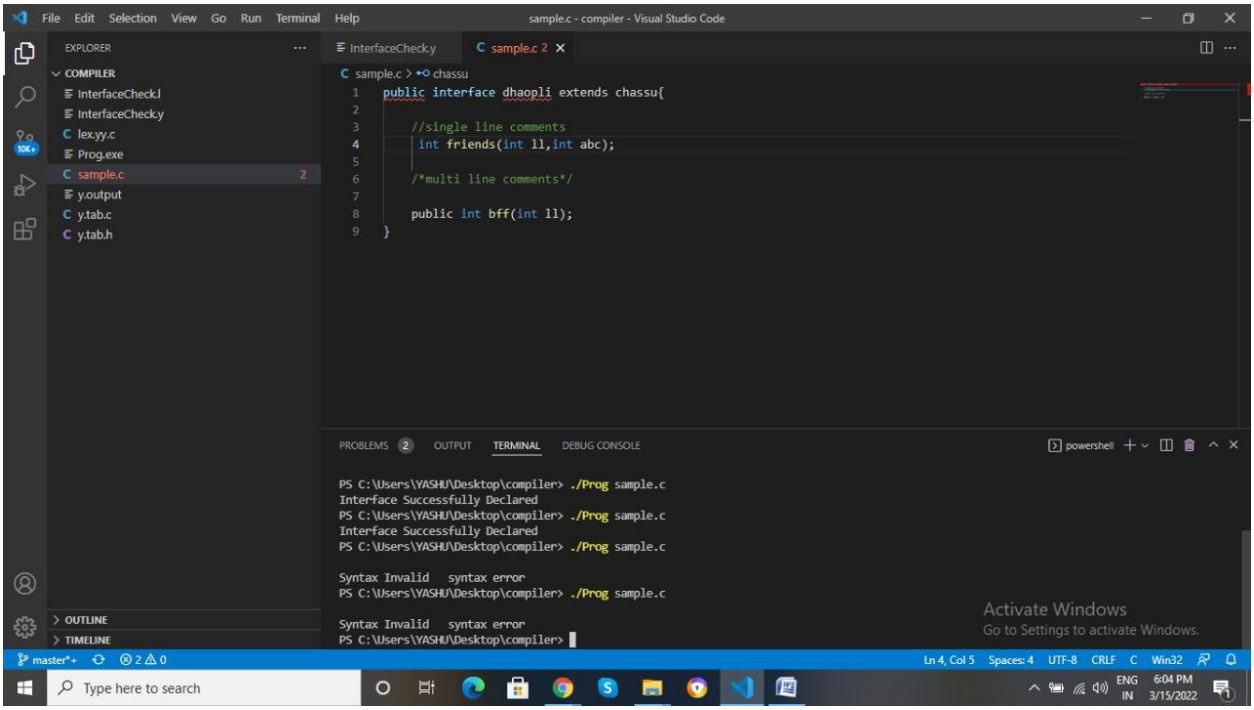
5. Handled if the return type declared is anything except int , char, string, float.



6. Handled if public keyword is not written before interface keyword.



7. Handled if we miss return data type.



8. Handled if we put “,” after function parameters.