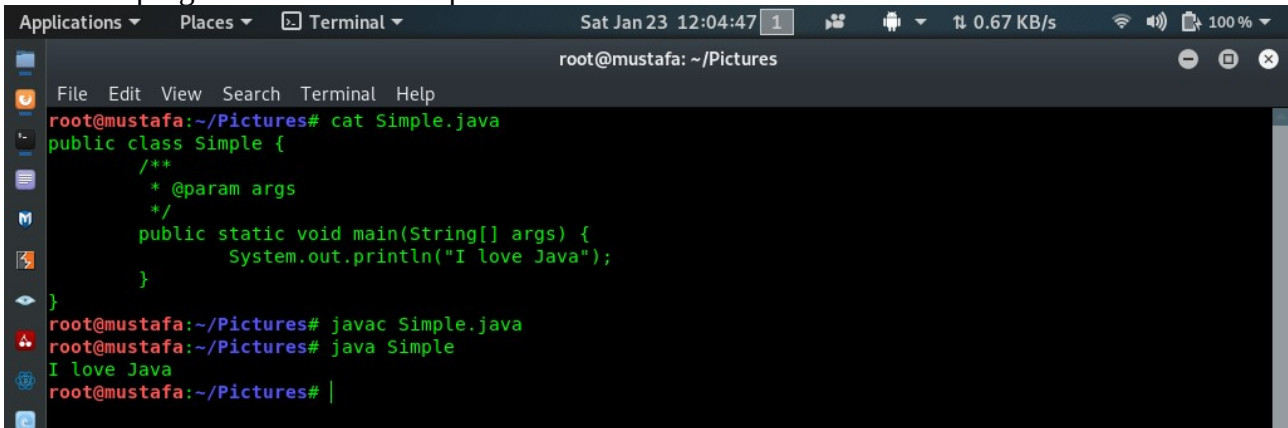


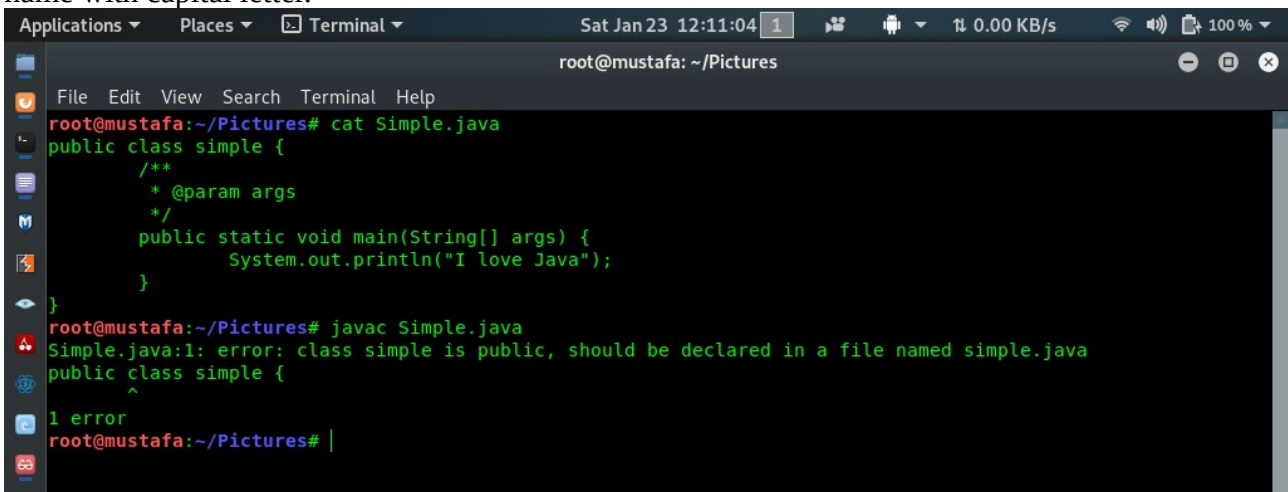
3. Solution For Program Names

When the program is run with Simple as an identifier



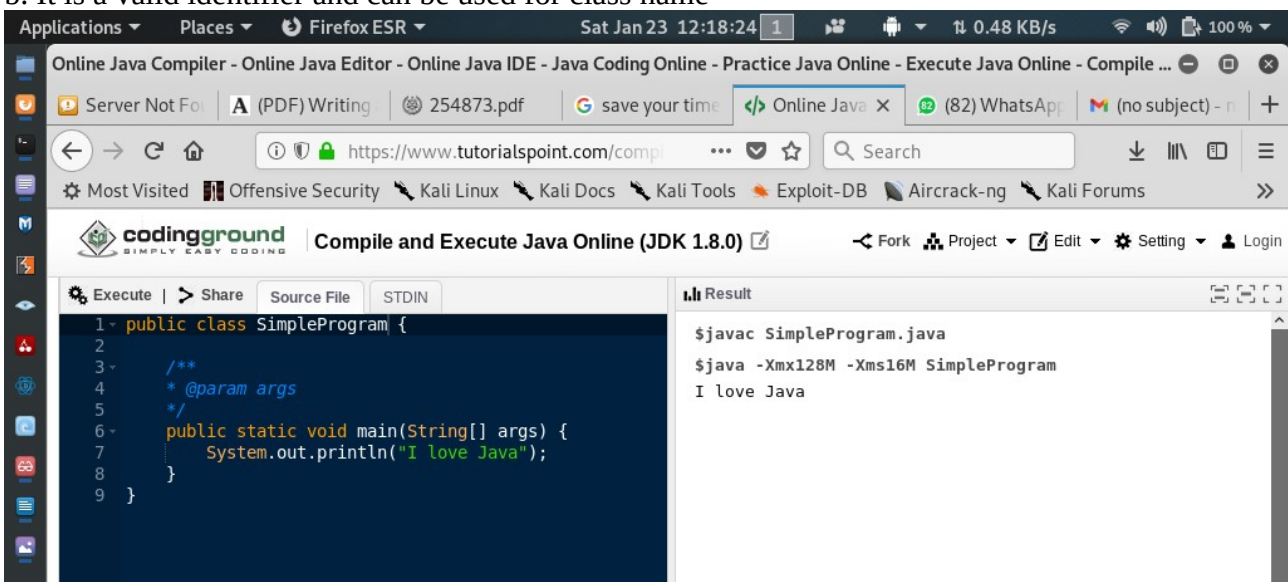
```
Applications ▾ Places ▾ Terminal ▾ Sat Jan 23 12:04:47 1 0.67 KB/s 100 %
root@mustafa: ~/Pictures
File Edit View Search Terminal Help
root@mustafa:~/Pictures# cat Simple.java
public class Simple {
    /**
     * @param args
     */
    public static void main(String[] args) {
        System.out.println("I love Java");
    }
}
root@mustafa:~/Pictures# javac Simple.java
root@mustafa:~/Pictures# java Simple
I love Java
root@mustafa:~/Pictures#
```

a. Java will not accept it as a class name and java file name since its supposed to start the class name with capital letter.



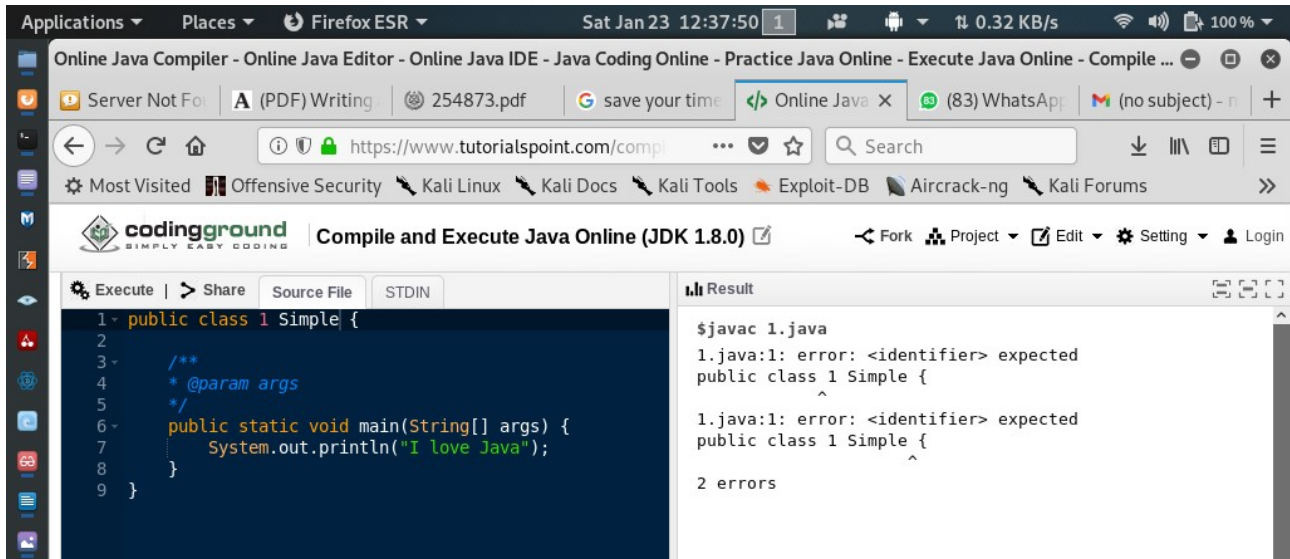
```
Applications ▾ Places ▾ Terminal ▾ Sat Jan 23 12:11:04 1 0.00 KB/s 100 %
root@mustafa: ~/Pictures
File Edit View Search Terminal Help
root@mustafa:~/Pictures# cat Simple.java
public class simple {
    /**
     * @param args
     */
    public static void main(String[] args) {
        System.out.println("I love Java");
    }
}
root@mustafa:~/Pictures# javac Simple.java
Simple.java:1: error: class simple is public, should be declared in a file named simple.java
public class simple {
      ^
1 error
root@mustafa:~/Pictures#
```

b. It is a valid identifier and can be used for class name



```
Applications ▾ Places ▾ Firefox ESR ▾ Sat Jan 23 12:18:24 1 0.48 KB/s 100 %
Online Java Compiler - Online Java Editor - Online Java IDE - Java Coding Online - Practice Java Online - Execute Java Online - Compile ...
Server Not For (PDF) Writing 254873.pdf save your time Online Java X (82) WhatsApp (no subject) - n
https://www.tutorialspoint.com/compi Search
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Kali Forums
codingground SIMPLY EASY CODING Compile and Execute Java Online (JDK 1.8.0) Fork Project Edit Setting Login
Execute | Share Source File STDIN Result
1- public class SimpleProgram {
2
3- /**
4-  * @param args
5-  */
6- public static void main(String[] args) {
7-     System.out.println("I love Java");
8- }
9- }
$javac SimpleProgram.java
$java -Xmx128M -Xms16M SimpleProgram
I love Java
```

c. It is an invalid identifier since in java, identifier shouldn't start with a number and a space is not allowed.



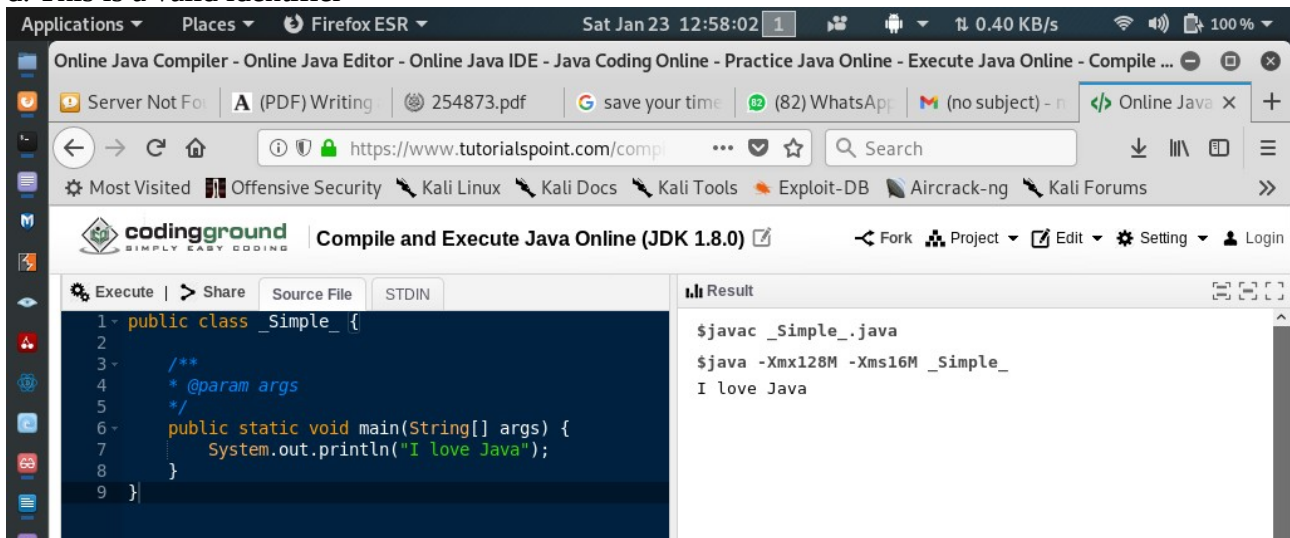
The screenshot shows the 'codingground' online Java compiler interface. The source code in the editor is:

```
1- public class 1 Simple {  
2  
3- /**  
4-  * @param args  
5-  */  
6- public static void main(String[] args) {  
7-     System.out.println("I love Java");  
8- }  
9- }
```

The 'Result' pane on the right shows the compilation output:

```
$javac 1.java  
1.java:1: error: <identifier> expected  
public class 1 Simple {  
           ^  
1.java:1: error: <identifier> expected  
public class 1 Simple {  
           ^  
2 errors
```

d. This is a valid identifier



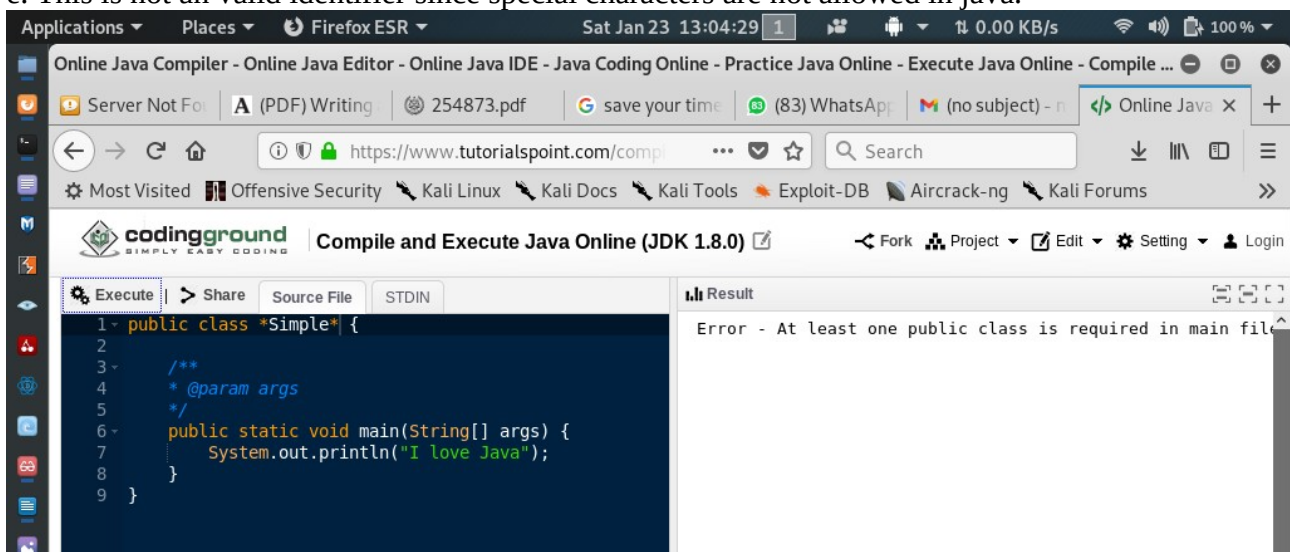
The screenshot shows the 'codingground' online Java compiler interface. The source code in the editor is:

```
1- public class _Simple_ {  
2  
3- /**  
4-  * @param args  
5-  */  
6- public static void main(String[] args) {  
7-     System.out.println("I love Java");  
8- }  
9- }
```

The 'Result' pane on the right shows the compilation output:

```
$javac _Simple_.java  
$java -Xmx128M -Xms16M _Simple_  
I love Java
```

e. This is not an valid identifier since special characters are not allowed in java.



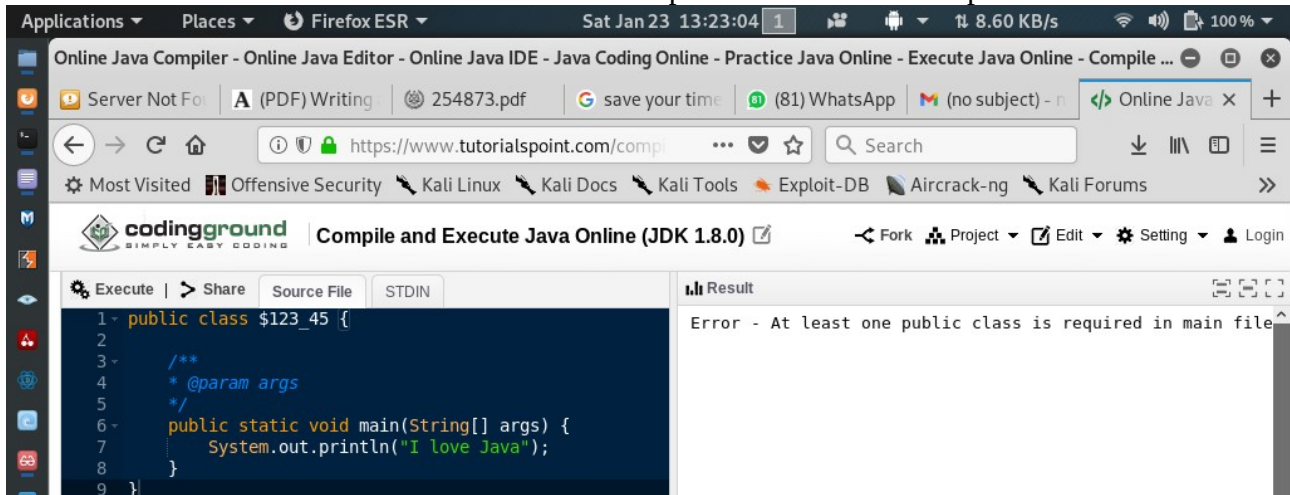
The screenshot shows the 'codingground' online Java compiler interface. The source code in the editor is:

```
1- public class *Simple* {  
2  
3- /**  
4-  * @param args  
5-  */  
6- public static void main(String[] args) {  
7-     System.out.println("I love Java");  
8- }  
9- }
```

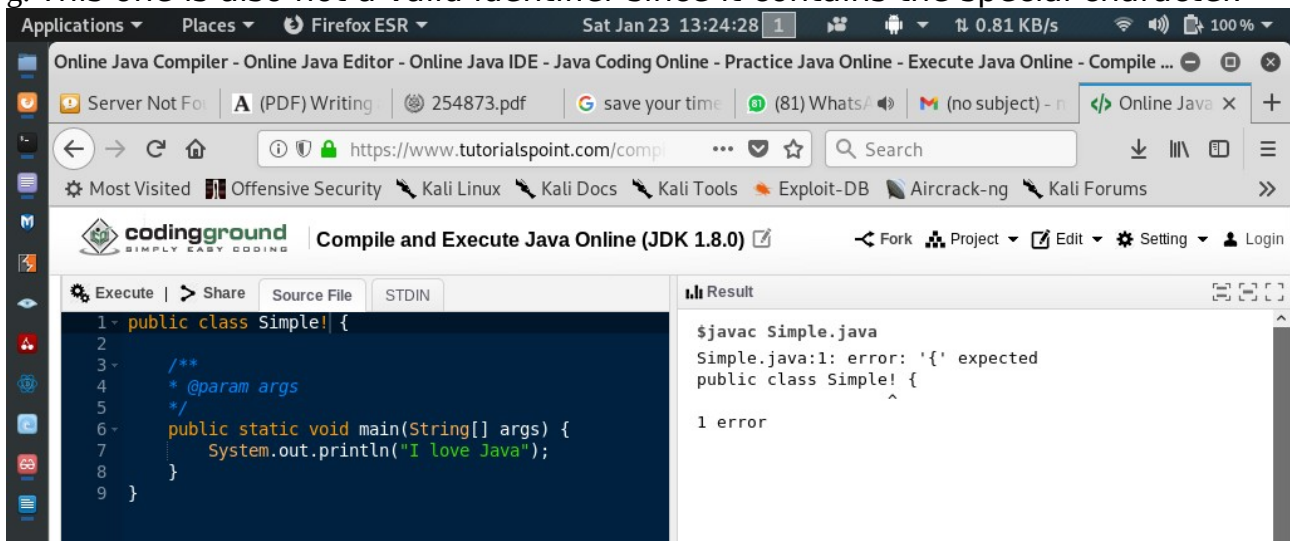
The 'Result' pane on the right shows the compilation output:

```
Error - At least one public class is required in main file
```

f. This is not a valid identifier since it contains the special character and a space



g. This one is also not a valid identifier since it contains the special character.



4. Solution For Recognizing Syntax Error

a. Program

```
public class Helo  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Helo World!");  
    }  
}
```

Output: It returns the error as; because filename and classname are different

Hello.java:1: class Helo is public, should be declared in a file named Helo.java
public class Hello

b. Misspelling inside string

```
public class Hello
{
public static void main(String[] args)
{
System.out.println("Helo World!");
}
}
```

OUTPUT will be displayed as

Helo World!

Because, the string within the quotation will be displayed as it is

c. No ending quotation mark in a string literal

```
public class Hello
{
public static void main(String[] args)
{
System.out.println("Hello World!);
}
}
```

OUTPUT will be displayed with the syntax error because there is no ending quotation mark

```
//Hello.java:5: error: unclosed string literal
System.out.println("Hello World!);
^
/Hello.java:5: error: ';' expected
System.out.println("Hello World!);
^
/Hello.java:7: error: reached end of file while parsing
}
^
3 errors
```

d. No beginning quotation mark in a string literal

```
public class Hello
{
public static void main(String[] args)
{
System.out.println(Hello World!");
}
}
```

OUTPUT will be displayed with the syntax error, because there is no beginning quotation mark

```
/Hello.java:5: error: ')' expected
System.out.println(Hello World!");
^
/Hello.java:5: error: unclosed string literal
System.out.println(Hello World!");
^
/Hello.java:5: error: not a statement
System.out.println(Hello World!");
^
3 errors
```

e. No semicolon after a statement

```
public class Hello
{
public static void main(String[] args)
{
System.out.println("Hello World!")
}
}OUTPUT will be displayed with the syntax error, because there is no semicolon at the end of the
statement
```

```
/Hello.java:5: error: ';' expected
System.out.println("Hello World!")
^
1 error
```

5. Solution for Corresponding Syntax Error

// Person.java

```
public class Person {
    public int age=12;

}
```

=====

// ProtectedClass.java

```
package packageOne;

public class ProtectedClass {
    protected long id = 1L;
```

```
protected void printID() {  
    System.out.println(this.id);  
}  
}
```

=====

```
// ProtectedSubClass.java
```

```
package packageTwo;
```

```
import packageOne.ProtectedClass;
```

```
public class ProtectedSubClass extends ProtectedClass {  
  
}
```

=====

```
// AccessModifiers.java
```

```
package packageOne;
```

```
import packageTwo.ProtectedSubClass;
```

```
public class AccessModifiers {
```

```
    public static void main(String[] args) {
```

```
        Person adam=new Person();
```

```
        adam.age=15;
```

```
        System.out.println(adam.age);
```

```
        ProtectedSubClass psc=new ProtectedSubClass();
```

```
        psc.printID();
```

```
    }
```

```
}
```

=====

output:

15

1

6. Solution for Two Meanings of +

The output is :

This is a long string that is the concatenation of two shorter strings

The first computer was invented about 55 years ago

8 plus 5 is 85

8 plus 5 is 13

85 equals 8 plus 5

In the first output line + acts as a concatenation operator

In 2nd line also as concatenation operator

In 3rd line also as concatenation operator for 3 operands

In 4th line + inside brackets act as addition operator evaluates as expression and outside one acts as concatenation

In 5th line + acts as concatenation only.