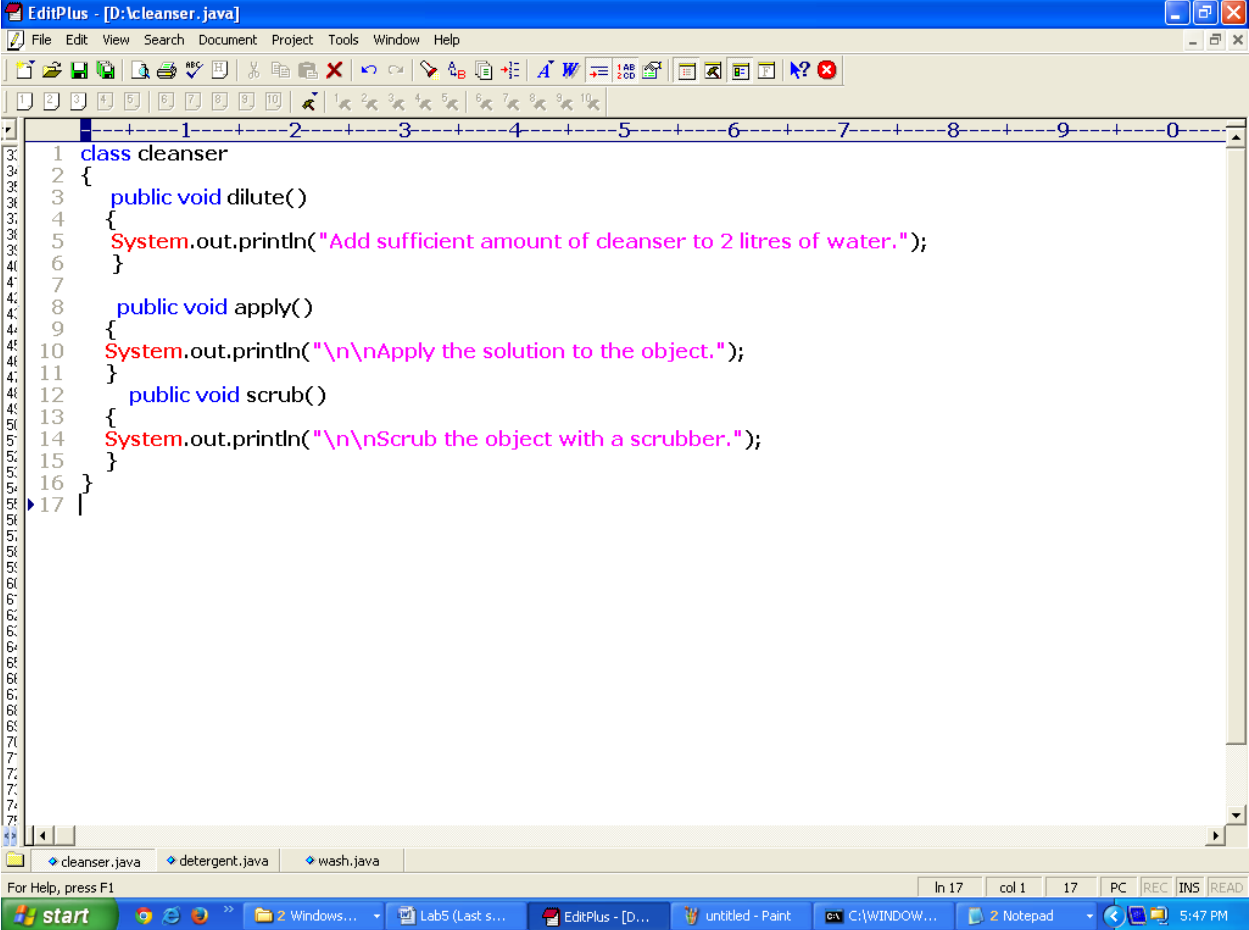


# Lab-5 Assignments

Introducing Inheritance (is a type of relationship)

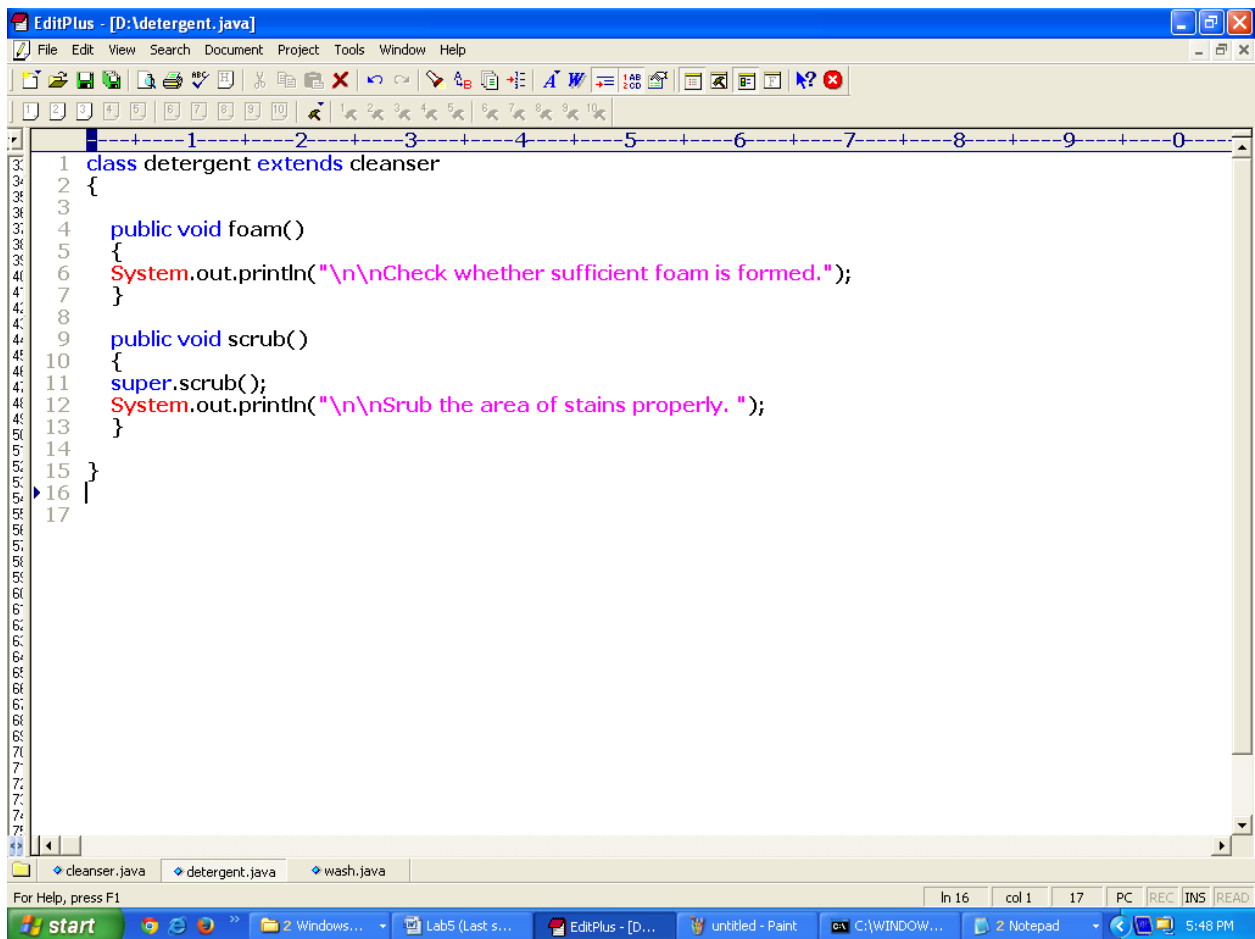
Demo for Inheritance



The screenshot shows the EditPlus IDE with a file named 'cleanser.java' open. The code defines a class 'cleanser' with three methods: 'dilute()', 'apply()', and 'scrub()'. Each method prints a message to the console. The IDE interface includes a menu bar (File, Edit, View, Search, Document, Project, Tools, Window, Help), a toolbar with various editing tools, and a status bar at the bottom showing the current line (17) and column (1).

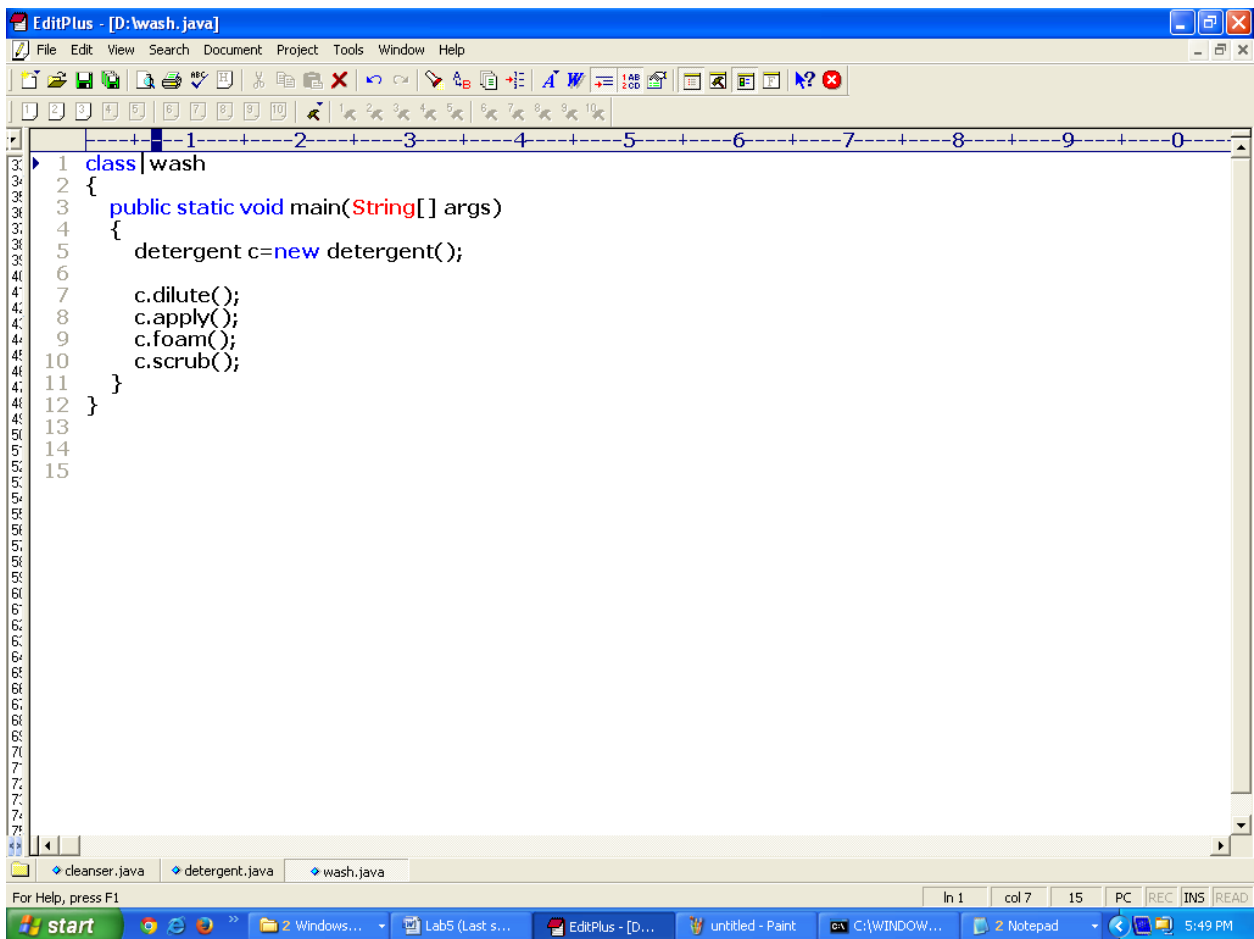
```
1 class cleanser
2 {
3     public void dilute()
4     {
5         System.out.println("Add sufficient amount of cleanser to 2 litres of water.");
6     }
7
8     public void apply()
9     {
10        System.out.println("\n\nApply the solution to the object.");
11    }
12    public void scrub()
13    {
14        System.out.println("\n\nScrub the object with a scrubber.");
15    }
16 }
17
```

Save as cleanser.java



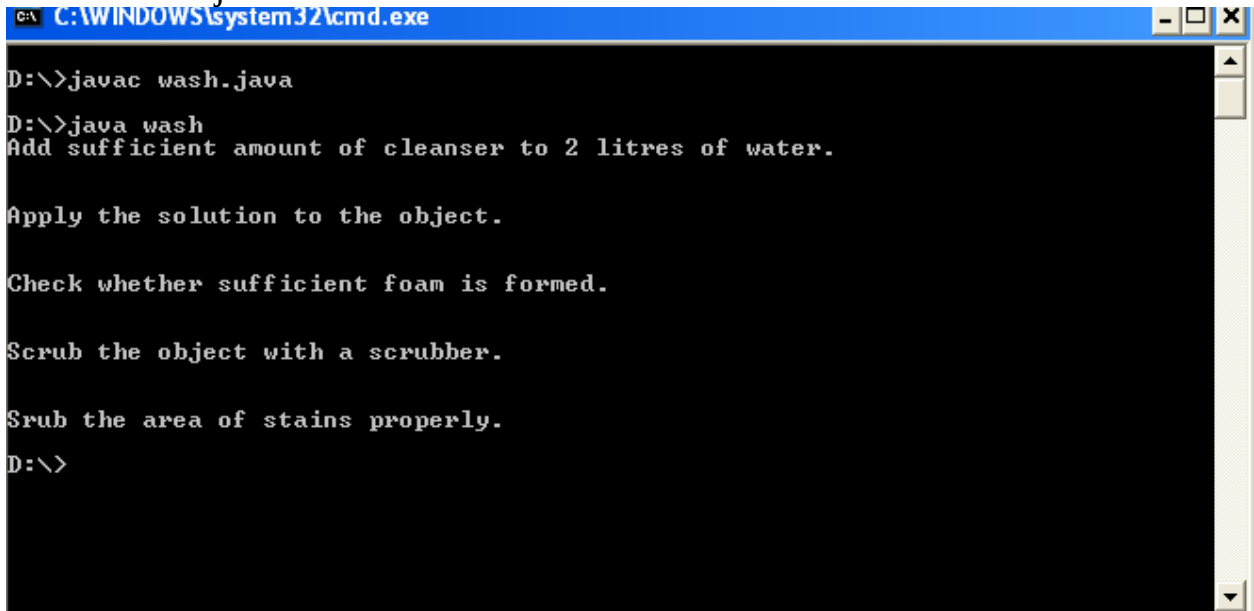
```
1 class detergent extends cleanser
2 {
3
4     public void foam()
5     {
6         System.out.println("\n\nCheck whether sufficient foam is formed.");
7     }
8
9     public void scrub()
10    {
11        super.scrub();
12        System.out.println("\n\nScrub the area of stains properly. ");
13    }
14
15 }
16
17
```

Save as detergent.java



```
1 class wash
2 {
3     public static void main(String[] args)
4     {
5         detergent c=new detergent();
6
7         c.dilute();
8         c.apply();
9         c.foam();
10        c.scrub();
11    }
12 }
13
14
15
```

Save as wash.java



```
C:\WINDOWS\system32\cmd.exe

D:\>javac wash.java

D:\>java wash
Add sufficient amount of cleanser to 2 litres of water.

Apply the solution to the object.

Check whether sufficient foam is formed.

Scrub the object with a scrubber.

Scrub the area of stains properly.
D:\>
```

Demo2

```
1 class A
2 {
3     A()
4     {System.out.println("\n\nInside A's default constructor. ");}
5 }
6
7 class B
8 {
9     B()
10    {System.out.println("\n\nInside B's default constructor. ");}
11 }
12
13
14 class C extends A
15 {
16     B b1=new B();
17
18     public static void main(String args[])
19     {
20
21         C c1=new C();
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
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
```

Save as C.java

```
D:\>javac C.java
D:\>java C

Inside A's default constructor.

Inside B's default constructor.
D:\>
```

## Assignments To Solve

1. Write class game{ }

```
class boardgame extends game{ }
class chess extends boardgame{ }
class gamedemo{
    public static void main(String[] args)    {
        chess ch=new chess();    }
}
```

(Note: Write only default constructors in each class with specific information . See the order of constructors invoked when chess object is created)

2. Create class WageEmployee extending Employee class with attributes as hrs (int) and rate(int) and method computeSalary() to calculate the salary. Print the salary and details of WageEmployee.  
(Note: Use the previous date and Employee classes. Accept the values from the user..Default, Parameterised Constructor and toString() to be written in all the classes)
3. Create SalesPerson class extending WageEmployee with attributes as sales(int) and commission (int). Override the ComputeSalary() in Salesperson class and print the salary and details of SalesPerson
4. Create Manager class extending Employee class with attributes as fixedsalary(int) and incentives(int) and method computeSalary() to calculate the salary of Manager .Print the salary and details of Manager
5. Write a TestEmployee class to print the details of all types of employees (use array[] of Employee class)