**Rajan Gautam**

**19BCP101**

**Div. II, CE 19**

**SOT, PDPU**

**Pandit Deendayal Energy University**

**School of Technology**

**Design Pattern and Thinking (20CP210P)**

**B. Tech - Computer Science & Engineering (Sem-IV)**

**Table of Contents**

[**Lab 1 Assignment: Factory Design Pattern for Creation of OS for Phones 1**](#_Toc63695678)

[**AIM: To write a Java Program to implement Factory Design Pattern for Creation of OS for Phones 1**](#_Toc63695679)

[**CODE: 1**](#_Toc63695680)

[**MainFactory.java 1**](#_Toc63695681)

[**OS\_Factory.java 2**](#_Toc63695682)

[**OS.java 2**](#_Toc63695683)

[**Android.java 3**](#_Toc63695684)

[**IOS.java 4**](#_Toc63695685)

[**Windows.java 5**](#_Toc63695686)

[**KaiOS.java 6**](#_Toc63695687)

[**Tizen.java 7**](#_Toc63695688)

[**OUTPUT: 9**](#_Toc63695689)

# Lab 1 Assignment: Factory Design Pattern for Creation of OS for Phones

## AIM: To write a Java Program to implement Factory Design Pattern for Creation of OS for Phones

## CODE:

### MainFactory.java

1. **package Phone\_OS;**
3. **import java.util.Scanner;**
5. **public class MainFactory {**
7. **public static void main(String[] args) {**
8. **OS\_Factory myOS = new OS\_Factory();**
9. **Scanner scanner = new Scanner(System.in);**
11. **System.out.println("<--- Select by Developer --->");**
12. **System.out.println("1. Apple");**
13. **System.out.println("2. Google");**
14. **System.out.println("3. Microsoft");**
15. **System.out.println("4. KaiOS");**
16. **System.out.println("5. Samsung");**
17. **System.out.print("Enter your choice: ");**
19. **String choice = scanner.nextLine();**
20. **scanner.close();**
22. **OS obj = myOS.getInstance(choice);**
24. **obj.name();**
25. **obj.specs();**
26. **obj.secure();**
27. **obj.developer();**
28. **obj.messaging();**
29. **obj.source();**
30. **obj.written\_language();**
31. **obj.kernel\_type();**
32. **obj.working\_state();**
34. **}**
36. **}**

### OS\_Factory.java

1. **package Phone\_OS;**
3. **public class OS\_Factory {**
4. **public OS getInstance(String str)**
5. **{**
6. **if (str.equals("Google")) {**
7. **return new Android();**
8. **}**
10. **if (str.equals("Apple")) {**
11. **return new IOS();**
12. **}**
14. **if (str.equals("KaiOS")) {**
15. **return new KaiOS();**
16. **}**
18. **if (str.equals("Microsoft")) {**
19. **return new Windows();**
20. **}**
22. **if (str.equals("Samsung")) {**
23. **return new Tizen();**
24. **}**
26. **return null;**
27. **}**
28. **}**

### OS.java

1. **package Phone\_OS;**
3. **public interface OS**
4. **{**
5. **public void name();**
6. **public void specs();**
7. **public void secure();**
8. **public void source();**
9. **public void developer();**
10. **public void messaging();**
11. **public void written\_language();**
12. **public void kernel\_type();**
13. **public void working\_state();**
14. **}**

### Android.java

1. **package Phone\_OS;**
3. **public class Android implements OS**
4. **{**
5. **@Override**
6. **public void name() {**
7. **System.out.println("<--Android-->");**
8. **}**
10. **@Override**
11. **public void specs() {**
12. **System.out.println("Most used phone OS.");**
13. **}**
15. **@Override**
16. **public void secure() {**
17. **System.out.println("Moderately Secured.");**
19. **}**
21. **@Override**
22. **public void source() {**
23. **System.out.println("Source Model: Open Source.");**
25. **}**
27. **@Override**
28. **public void developer() {**
29. **System.out.println("Product of Google.");**
31. **}**
33. **@Override**
34. **public void messaging() {**
35. **System.out.println("Provides Message app for messaging.");**
37. **}**
39. **@Override**
40. **public void written\_language() {**
41. **System.out.println("Written Language: Java, C, C++.");**
43. **}**
45. **@Override**
46. **public void kernel\_type() {**
47. **System.out.println("Kernel Type: Linux");**
49. **}**
51. **@Override**
52. **public void working\_state() {**
53. **System.out.println("Working State: Current");**
55. **}**
56. **}**

### IOS.java

1. **package Phone\_OS;**
3. **public class IOS implements OS**
4. **{**
5. **@Override**
6. **public void name() {**
7. **System.out.println("<--IOS-->");**
8. **}**
10. **@Override**
11. **public void specs()**
12. **{**
13. **System.out.println("Most secure phone OS.");**
14. **}**
16. **@Override**
17. **public void secure() {**
18. **System.out.println("Most Secured Phone.");**
20. **}**
21. **@Override**
22. **public void source() {**
23. **System.out.println("Source Model: Closed Source.");**
25. **}**
27. **@Override**
28. **public void developer() {**
29. **System.out.println("Product of Apple.");**
31. **}**
33. **@Override**
34. **public void messaging() {**
35. **System.out.println("Provides iMessage for messaging.");**
37. **}**
39. **@Override**
40. **public void written\_language() {**
41. **System.out.println("Written Language: C, C++, Swift.");**
43. **}**
45. **@Override**
46. **public void kernel\_type() {**
47. **System.out.println("Kernel Type: Hybrid");**
49. **}**
51. **@Override**
52. **public void working\_state() {**
53. **System.out.println("Working State: Current");**
55. **}**
56. **}**

### 

### Windows.java

1. **package Phone\_OS;**
3. **public class Windows implements OS**
4. **{**
5. **@Override**
6. **public void name() {**
7. **System.out.println("<--Windows-->");**
8. **}**
10. **@Override**
11. **public void specs() {**
12. **System.out.println("I am about to die.");**
13. **}**
15. **@Override**
16. **public void secure() {**
17. **System.out.println("Not so Secured.");**
19. **}**
20. **@Override**
21. **public void source() {**
22. **System.out.println("Source Model: Closed Source.");**
24. **}**
26. **@Override**
27. **public void developer() {**
28. **System.out.println("Product of Microsoft.");**
30. **}**
32. **@Override**
33. **public void messaging() {**
34. **System.out.println("User needs to download separate app for messaging.");**
36. **}**
38. **@Override**
39. **public void written\_language() {**
40. **System.out.println("Written Language: C, C++.");**
42. **}**
44. **@Override**
45. **public void kernel\_type() {**
46. **System.out.println("Kernel Type: Hybrid.");**
48. **}**
50. **@Override**
51. **public void working\_state() {**
52. **System.out.println("Working State: Discontinued");**
54. **}**
55. **}**

### 

### KaiOS.java

1. **package Phone\_OS;**
3. **public class KaiOS implements OS**
4. **{**
5. **@Override**
6. **public void name() {**
7. **System.out.println("<--KaiOS-->");**
8. **}**
10. **@Override**
11. **public void specs()**
12. **{**
13. **System.out.println("Emerging OS.");**
14. **}**
16. **@Override**
17. **public void secure() {**
18. **System.out.println("Moderately Secured OS.");**
20. **}**
21. **@Override**
22. **public void source() {**
23. **System.out.println("Source Model: Open Source.");**
25. **}**
27. **@Override**
28. **public void developer() {**
29. **System.out.println("Product of KaiOS Technologies.");**
31. **}**
33. **@Override**
34. **public void messaging() {**
35. **System.out.println("Provide inbuilt messaging app.");**
37. **}**
39. **@Override**
40. **public void written\_language() {**
41. **System.out.println("Written Language: HTML, CSS, JavaScript.");**
43. **}**
45. **@Override**
46. **public void kernel\_type() {**
47. **System.out.println("Kernel Type: Monolithic");**
49. **}**
51. **@Override**
52. **public void working\_state() {**
53. **System.out.println("Working State: Current");**
55. **}**
56. **}**

### 

### Tizen.java

1. **package Phone\_OS;**
3. **public class Tizen implements OS**
4. **{**
5. **@Override**
6. **public void name() {**
7. **System.out.println("<--Tizen-->");**
8. **}**
10. **@Override**
11. **public void specs() {**
12. **System.out.println("Electronic Devices' OS.");**
14. **}**
16. **@Override**
17. **public void secure() {**
18. **System.out.println("Moderately Secured.");**
20. **}**
22. **@Override**
23. **public void source() {**
24. **System.out.println("Source Model: Open Source.");**
26. **}**
28. **@Override**
29. **public void developer() {**
30. **System.out.println("Product of Samsung.");**
32. **}**
34. **@Override**
35. **public void messaging() {**
36. **System.out.println("Provides Messages app for messaging.");**
38. **}**
40. **@Override**
41. **public void written\_language() {**
42. **System.out.println("Written Language: HTML5, C, C++.");**
44. **}**
46. **@Override**
47. **public void kernel\_type() {**
48. **System.out.println("Kernel Type: Monolithic.");**
50. **}**
52. **@Override**
53. **public void working\_state() {**
54. **System.out.println("Working State: Current");**
56. **}**
57. **}**

## 

## OUTPUT:

<--- Select by Developer --->

1. Apple

2. Google

3. Microsoft

4. KaiOS

5. Samsung

Enter your choice: Microsoft

<--Windows-->

I am about to die.

Not so Secured.

Product of Microsoft.

User needs to download separate app for messaging.

Source Model: Closed Source.

Written Language: C, C++.

Kernel Type: Hybrid.

Working State: Discontinued

<--- Select by Developer --->

1. Apple

2. Google

3. Microsoft

4. KaiOS

5. Samsung

Enter your choice: Apple

<--IOS-->

Most secure phone OS.

Most Secured Phone.

Product of Apple.

Provides iMessage for messaging.

Source Model: Closed Source.

Written Language: C, C++, Swift.

Kernel Type: Hybrid

Working State: Current