

#### 47) Abstract Factory Code :

- **Window.java**

```
package Training;  
  
public interface Window{  
  
    }  

```

- **PMWindow.java**

```
package Training;  
  
public class PMWindow implements Window{  
  
    }  

```

- **MotifWindow.java**

```
package Training;  
  
public class MotifWindow implements Window{  
  
    }  

```

- **WidgetFactory.java**

```
package Training;

public interface WidgetFactory{
public Window createWindow();
public Window createScrollBar();
}
```

- **PMWidgetFactory.java**

```
package Training;

public class PMWidgetFactory {

public Window createWindow(){
PMWindow window = new PMWindow();
return window;
}
}
```

- **MotifWidgetFactory.java**

```
package Training;

public class MotifWidgetFactory {

public Window createWindow(){
MotifWindow window = new MotifWindow();
return window;
}
}
```

- ClientClass.java

```
package Training;

public class ClientClass{
public void buildWindow(WidgetFactory widgetFactory){
Window window = widgetFactory.createWindow();

}
}
```

- Main.java

```
package Training;

public class Main{
public static void main(String[] args){
ClientClass cc = new ClientClass();
WidgetFactory widgetFactory = null;

if(Platform.currentPlatform()=="Motif"){
widgetFactory = new MotifWidgetFactory();
} else {
widgetFactory = new PMWidgetFactory();
}
builder.buildWindow(widgetFactory);
}
}
```