



48024 Applications Programming

Dr Angela Huo



Contents

Pre: Open Question Board(<https://padlet.com/angelahuo/appsprog>)

- Study 7 Assessment
- Assignment 1
- Key points of Study 8
- Lab8 Preview

New Patterns and Syntax

- Required new patterns and syntax:

1. The Observer Pattern
2. Inner Classes
3. Anonymous Inner Classes
4. Lambda Expressions

Quiz

- What is class B an example of?

```
public class A {  
    private class B {  
    }  
}
```

Quiz

- What is this an example of?

```
new A() {  
    @Override public void foo(int x) {  
    }  
}
```

Anonymous and Inner Class

- An interface cannot be instantiated since it has no implementation:



```
new ProductObserver()
```

- However, you can provide the implementation while instantiating it:



```
new ProductObserver() {  
    @Override public void handleSale(double money) {  
        System.out.println("You paid $" + money);  
    }  
}
```

- Same as defining a class that implements the interface, then creating a new instance of that class.

Except the class has no name. Hence, it is “anonymous”

Quiz

- What is this?

```
x -> System.out.println(x);
```

- A lambda expression with a syntax error. What is the error?
- The semicolon. When are semicolons allowed?
- Inside a block:

```
x -> { System.out.println(x); }
```

Quiz

- What is this?

```
public interface EventHandler<X> {  
    void handle(X event);  
}
```


Registering an observer as an inner class

```
public class MyApplication extends Application {  
    private TextField usernameTf;  
    private PasswordField passwordTf;  
    @Override public void start(Stage stage) {  
        Button loginBtn = new Button("Login");  
        loginBtn.setOnAction(new LoginButtonHandler());  
        ...  
    }  
    private class LoginButtonHandler implements EventHandler<ActionEvent> {  
        @Override public void handle(ActionEvent event) {  
            if (checkPassword(usernameTf.getText(), passwordPf.getText()))  
                ...  
        }  
    }  
}
```

Registering as an anonymous inner class

```
public class MyApplication extends Application {  
    private TextField usernameTf;  
    private PasswordField passwordTf;  
    @Override public void start(Stage stage) {  
        Button loginBtn = new Button("Login");  
        loginBtn.setOnAction(new EventHandler<ActionEvent>() {  
            @Override public void handle(ActionEvent event) {  
                if (checkPassword(usernameTf.getText(), passwordPf.getText())  
                    ...  
            }  
        });  
        ...  
    }  
}
```

Question: Why are they called **anonymous**? Why are they called **inner**?"

Registering as a lambda expression

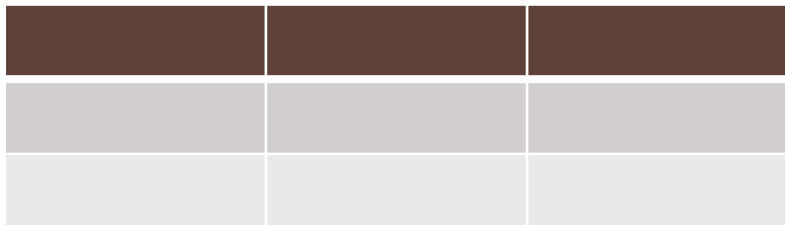
```
public class MyApplication extends Application {  
    private TextField usernameTf;  
    private PasswordField passwordTf;  
  
    @Override public void start(Stage stage) {  
        Button loginBtn = new Button("Login");  
        loginBtn.setOnAction(event -> {  
            if (checkPassword(usernameTf.getText(), passwordPf.getText()))  
                ...  
        });  
        ...  
    }  
}
```

Which one should I use?

- Use a lambda expression if the class has one method and is used once.
- Use an anonymous inner class if the class has multiple fields/methods.
- Use an inner class if you also need to create more than one instance.
- Use a normal class if you also need to access it from other classes(or if you anticipate needing to)

Question Board

- When to use GridPane, when using HBox?
 - GridPane(Table): Leaf nodes are displayed in matrix



- Hbox(List): Leaf nodes are horizontal displayed



Lab 8

- There's no entry for Lab 8. You need to access Lab 8 on Canvas.
- A quiz about Lab 8 programming process will be assessed on Canvas.
- 15-25min intro/demo
- Coding/Consultation - remaining time

- Topics: GUIs and events
 - inner classes
 - anonymous inner classes
 - lambda expression
 - These are the foundations for assignment 2

Question Board

- I can't import javafx, into my project, like the lab instructions say. It says the package does not exist.
 - Check if you use Java 8 for the project. If not, please go to Study 7 and watch the NetBeans installation demo.
- My event handlers keep getting errors. The message is “to create method in calculator. “
 - **Hint:** The LoginForm class should also be created in the default package alongside your existing Calculator class. In NetBeans, you can run the LoginForm class by right-clicking on it and selecting "Run File".

The coming weeks

- Lab 9 introduce MVC
- Assignment 2 released
- Other assignments due at the same time.....
- **Time management is very important!**
 - Watch the study modules carefully and write down the patterns in your pattern book.
 - Watch the lab demo video before the lab session to better prepare your questions to ask in the lab session.
 - Finish each lab no matter whether it is counted into your final grade, because every answer about the assignment 2 is already clearly stated in the study materials.

Contact

- Subject Coordinator and Lecturer: Angela Huo
- Email: huan.huo@uts.edu.au
- Contact information on Canvas

See you next week!