2025/11/19 Wednesday

- Quiz #4, next class, Friday (Nov. 21)

- HW 10, please check the key.
- HW 11, 24hours grace period
- HW 12

- Final Exam (Test 3). In class test only. NO online test!

Pearson

Review Java Swing

- You already learned basic Java GUI.

- Today, inner class and anonymous inner class

- Next class:
- Lambda expressions, animation, and quiz

- Four pillars of object-oriented programming (OOP) are
  abstraction,
  encapsulation,
  inheritance,
  and **polymorphism**

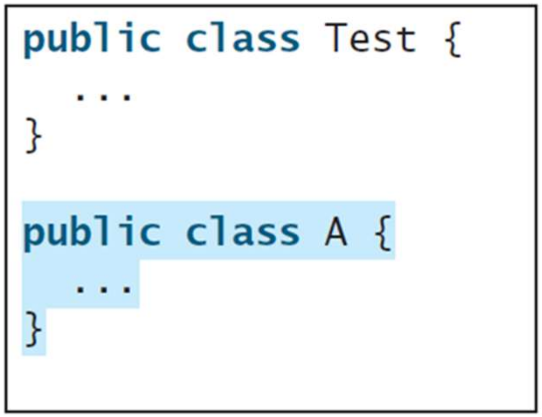
Inner Classes (1 of 4)

Inner class: A class is a member of another class.

Advantages: In some applications, you can use an inner class to make programs simple.

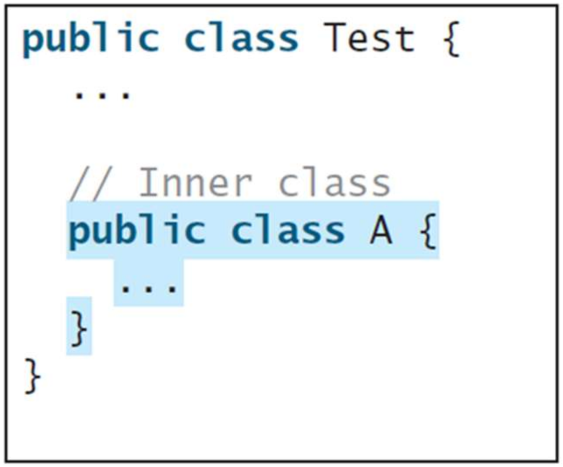
An inner class can reference the data and methods defined in the outer class in which it nests, so you do not need to pass the reference of the outer class to the constructor of the inner class.

Inner Classes (2 of 4)



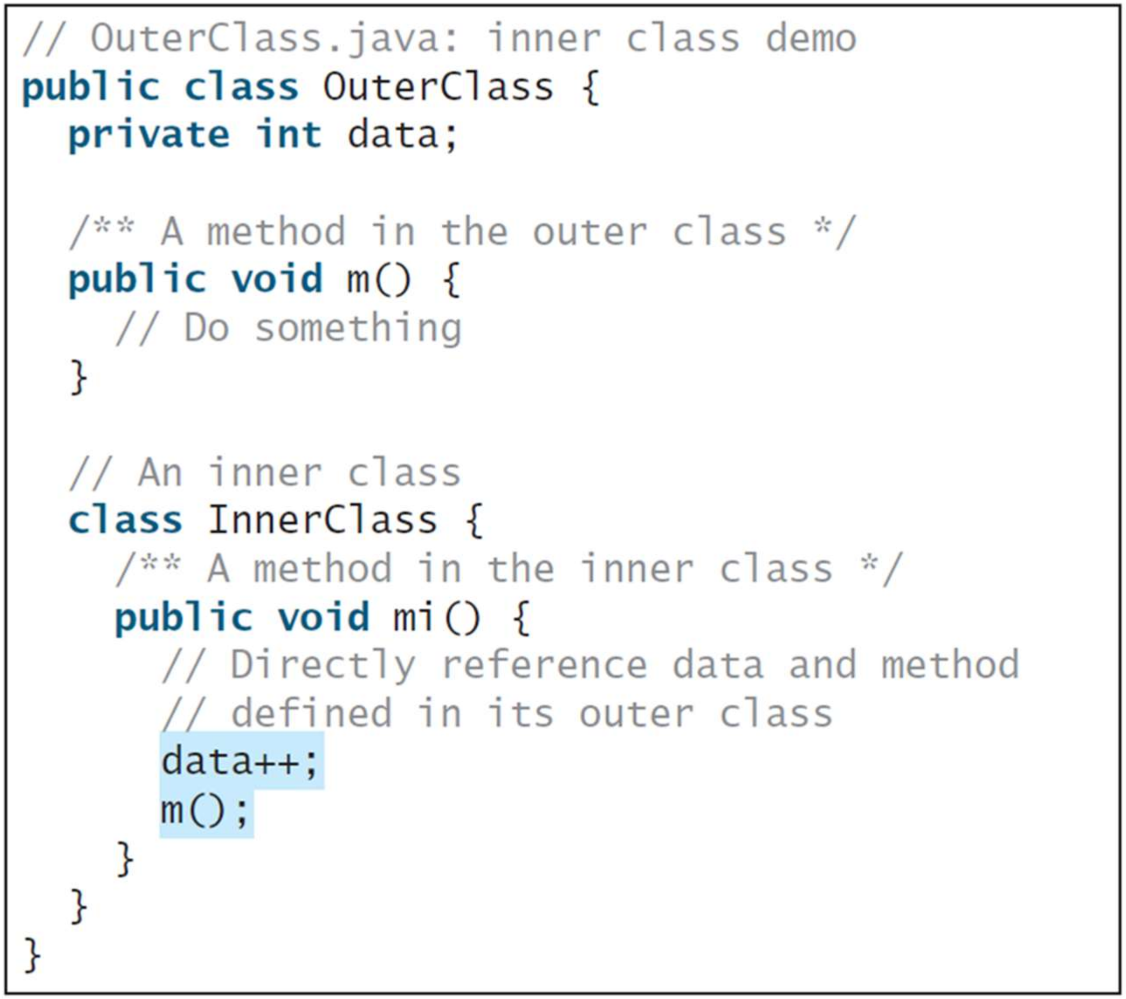
```
public class Test {
  ...
}

public class A {
  ...
}
```

(a)



```
public class Test {
  ...

  // Inner class
  public class A {
    ...
  }
}
```

(b)



```
// OuterClass.java: inner class demo
public class OuterClass {
  private int data;

  /** A method in the outer class */
  public void m() {
    // Do something
  }

  // An inner class
  class InnerClass {
    /** A method in the inner class */
    public void mi() {
      // Directly reference data and method
      // defined in its outer class
      data++;
      m();
    }
  }
}
```

(c)

Inner Classes (3 of 4)

Inner classes can make programs simple and concise.

An inner class supports the work of its containing outer class and is compiled into a class named **OuterClassName$InnerClassName**.class.

For example, the inner class InnerClass in OuterClass is compiled into **OuterClass$InnerClass.class**.

Inner Classes (4 of 4)

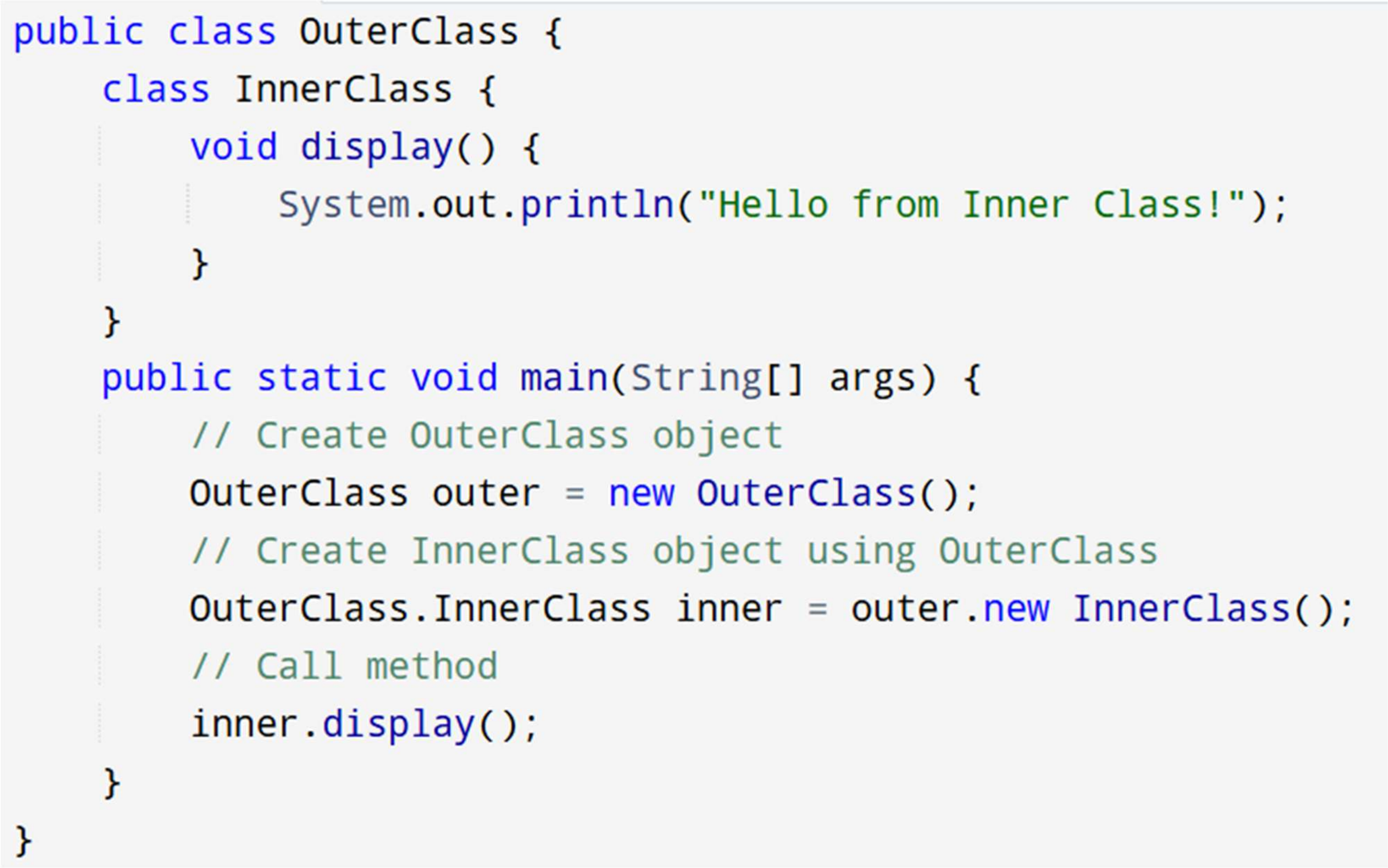
- An inner class can be declared public, protected, or private subject to the same visibility rules applied to a member of the class.
- An inner class can be declared static. A static inner class can be accessed using the outer class name. A static inner class cannot access nonstatic members of the outer class

Pearson

Practice One

Programiz

Online Java Compiler



```
public class OuterClass {
    class InnerClass {
        void display() {
            System.out.println("Hello from Inner Class!");
        }
    }
    public static void main(String[] args) {
        // Create OuterClass object
        OuterClass outer = new OuterClass();
        // Create InnerClass object using OuterClass
        OuterClass.InnerClass inner = outer.new InnerClass();
        // Call method
        inner.display();
    }
}
```

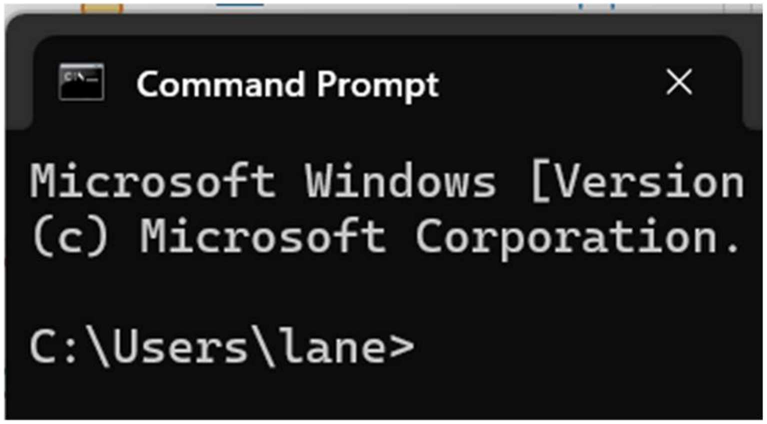
Pearson

Practice

- Now, with the command line interface
- Make/create a new folder for today

Lower case the first letter for each line below

- Notepad OuterClass.java
- Dir
- Javac OuterClass.java
- Dir
- Java OuterClass



```
Microsoft Windows [Version
(c) Microsoft Corporation.

C:\Users\lane>
```

Anonymous Inner Classes (1 of 3)

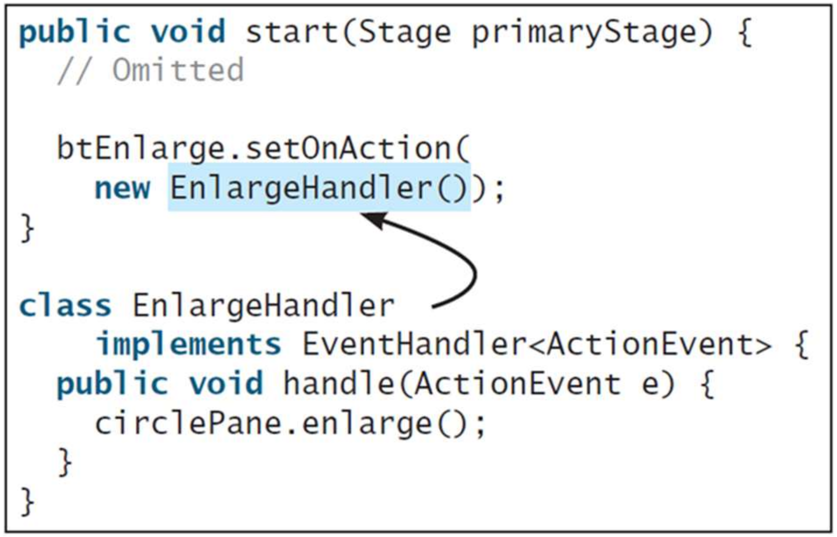
- An anonymous inner class must always extend a superclass or implement an interface, but it cannot have an explicit extends or implements clause.
- An anonymous inner class must implement all the abstract methods in the superclass or in the interface.
- An anonymous inner class always uses the no-arg constructor from its superclass to create an instance. If an anonymous inner class implements an interface, the constructor is Object().
- An anonymous inner class is compiled into a class named OuterClassName$n.class. For example, if the outer class Test has two anonymous inner classes, these two classes are compiled into Test$1.class and Test$2.class.

Anonymous Inner Classes (2 of 3)

An **anonymous inner class** is an inner class without a name. It combines declaring an inner class and creating an instance of the class in one step. An anonymous inner class is declared as follows:

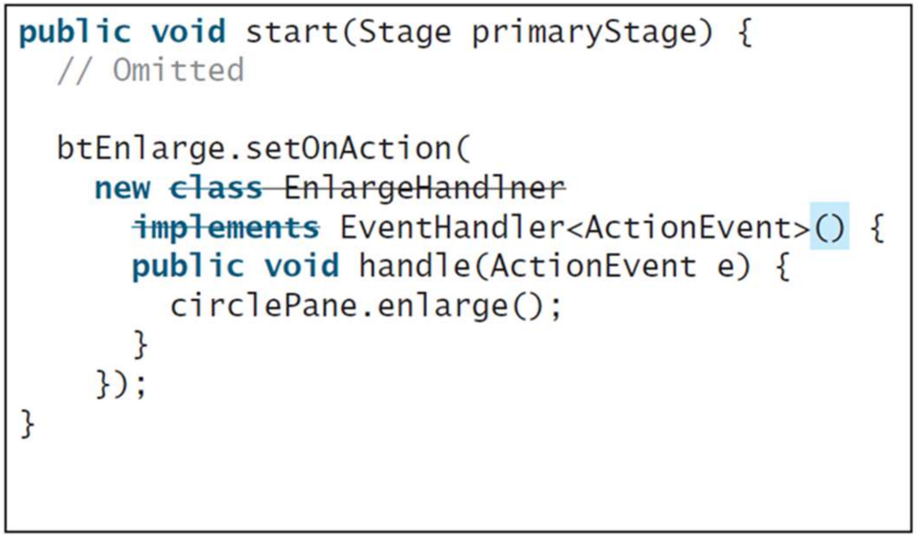
```
new SuperClassName/InterfaceName() {
  // Implement or override methods in superclass or
   interface
  // Other methods if necessary
}
```

Anonymous Inner Classes (3 of 3)



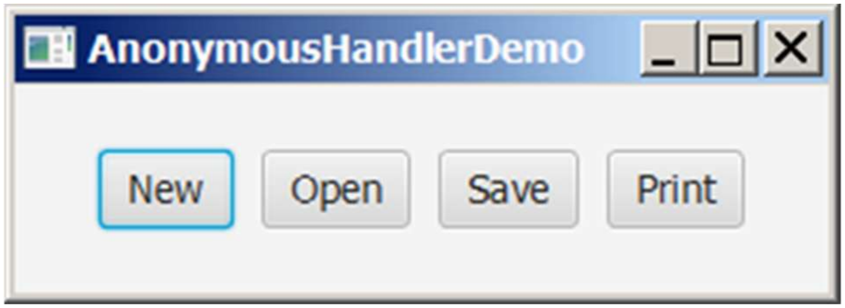
```
public void start(Stage primaryStage) {
  // Omitted

  btEnlarge.setOnAction(
    new EnlargeHandler());
}

class EnlargeHandler
    implements EventHandler<ActionEvent> {
  public void handle(ActionEvent e) {
    circlePane.enlarge();
  }
}
```

(a) Inner class EnlargeListener



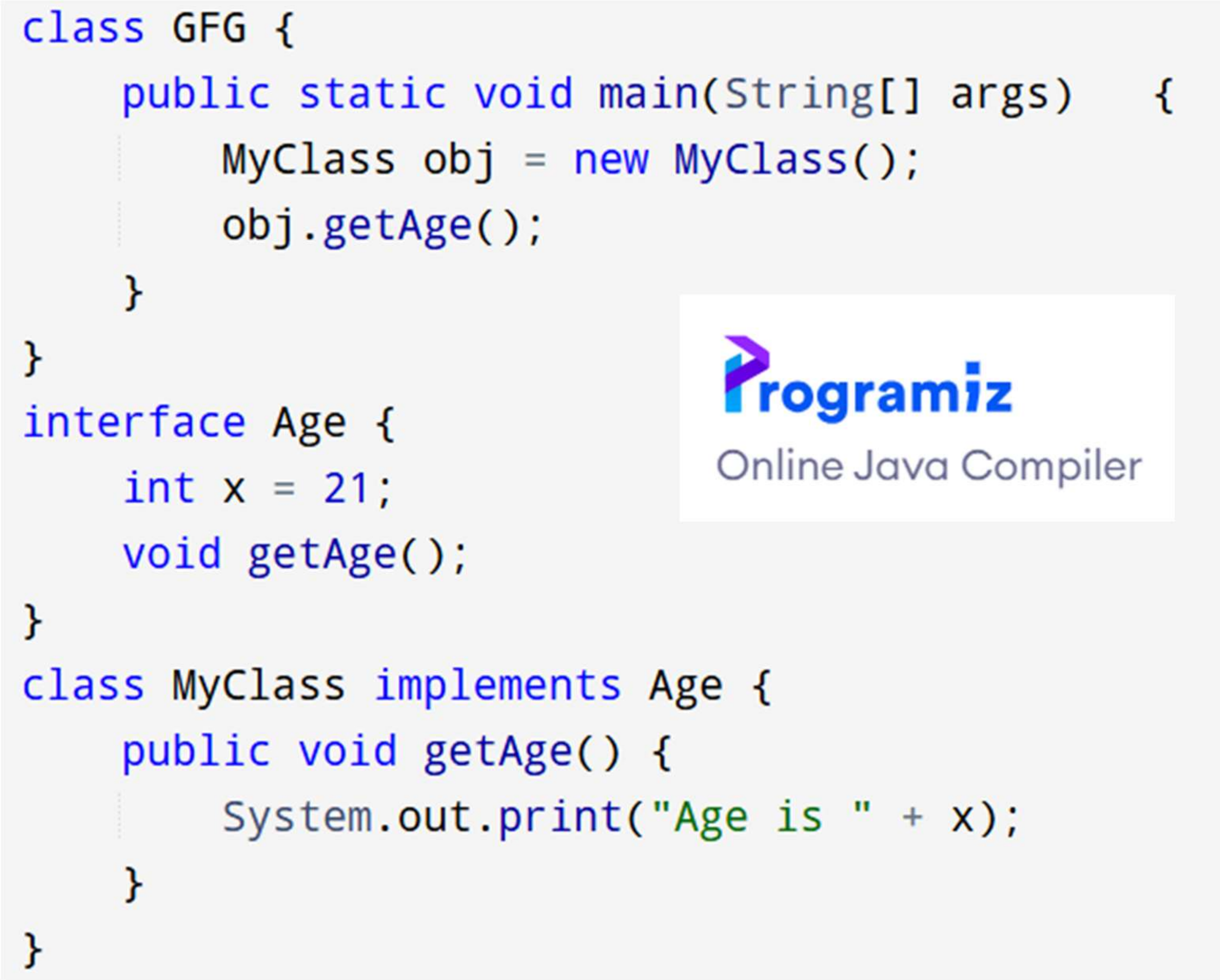
```
public void start(Stage primaryStage) {
  // Omitted

  btEnlarge.setOnAction(
    new ~~class EnlargeHandlner~~
      ~~implements~~ EventHandler<ActionEvent>() {
      public void handle(ActionEvent e) {
        circlePane.enlarge();
      }
    });
}
```

(b) Anonymous inner class



AnonymousHandlerDemo

Practice Two (from gfg)



```
class GFG {
    public static void main(String[] args)   {
        MyClass obj = new MyClass();
        obj.getAge();
    }
}
interface Age {
    int x = 21;
    void getAge();
}
class MyClass implements Age {
    public void getAge() {
        System.out.print("Age is " + x);
    }
}
```

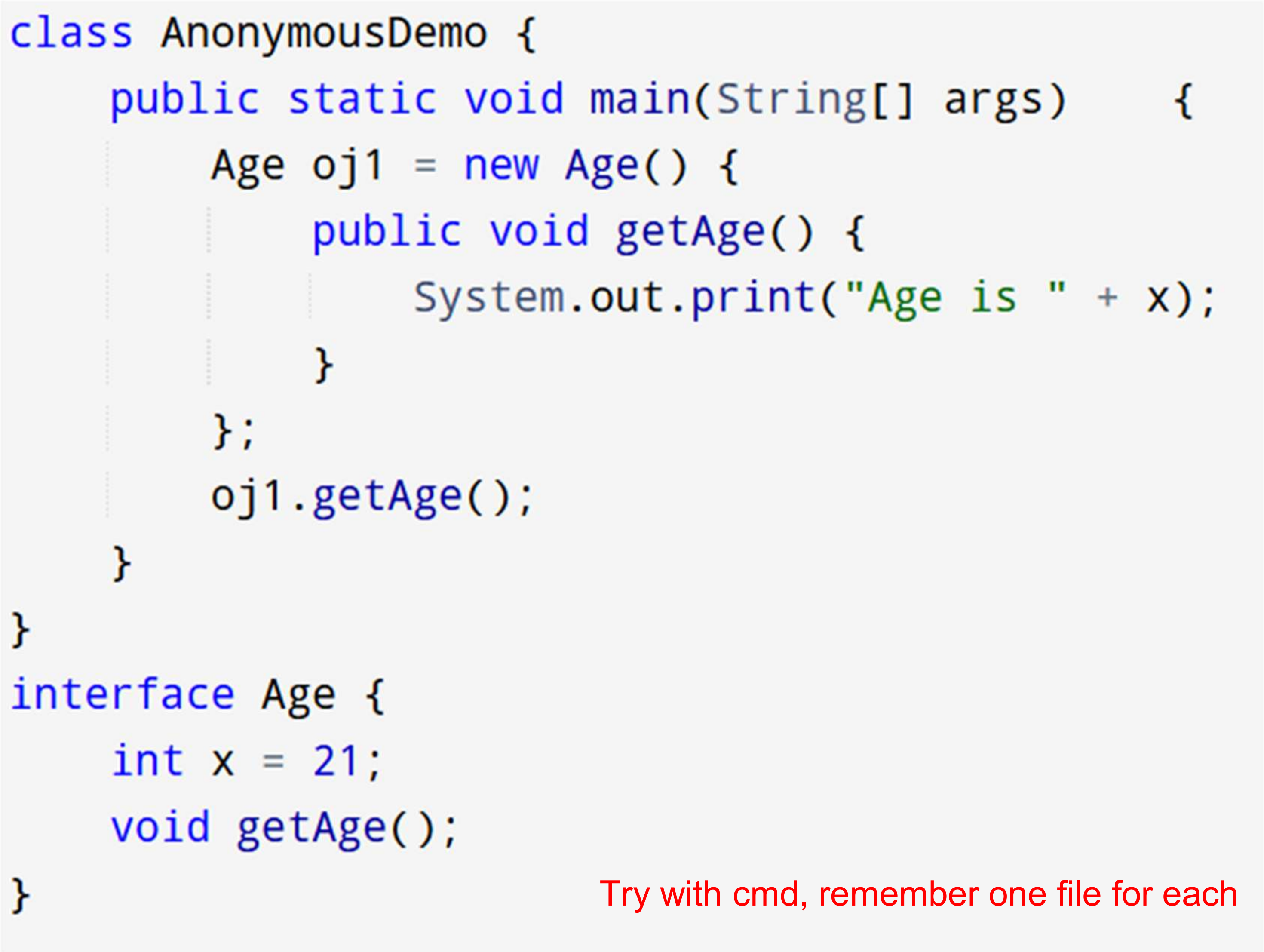
Programiz
Online Java Compiler

Pearson

Practice Three (from gfg)

Programiz

Online Java Compiler



```
class AnonymousDemo {
    public static void main(String[] args)    {
        Age oj1 = new Age() {
            public void getAge() {
                System.out.print("Age is " + x);
            }
        };
        oj1.getAge();
    }
}
interface Age {
    int x = 21;
    void getAge();
}
```

Try with cmd, remember one file for each

- Next class:
- Lambda expressions, animation, and quiz

- Final Exam (Test 3). In class test only. NO online test!