CHAPTER 2

■ ■ ■

Inner Classes

QUESTIONS AND EXERCISES

1. What is an inner class? Differentiate between member, local, and anonymous inner classes.

   Answer:

   An inner class is a class declared inside the body of another class. A member inner class is declared inside a class. It can be declared as `public`, `private`, `protected`, or package-level. A local inner class is declared inside a block. Its scope is limited to the block in which it is declared. An anonymous inner class is the same as a local inner class, but has no name. It is declared and an object of the class is created at the same time.

2. What is the fully qualified name of the inner class `B`, which is declared as follows?

```
// A.java
package com.jdojo.innerclasses.exercises;

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

   Answer:

   The fully qualified name of the inner class `B` is `com.jdojo.innerclasses.excercises.A.B`

3. Consider the following declaration for top-level class named `Cup` and a member inner class named `Handle`:

```
// Cup.java
package com.jdojo.innerclasses.exercises;

public class Cup {
   public class Handle {
       public Handle() {
          System.out.println("Created a handle for the cup");
       }
   }

   public Cup() {
       System.out.println("Created a cup");
   }
}
```

Complete the code in the `main()` method of the following `CupTest` class that will create an instance of the `Cup.Handle` inner class:

```
// CupTest.java
package com.jdojo.innerclasses.exercises;

public class CupTest {
   public static void main(String[] args) {
       // Create a Cup
       Cup c = new Cup();

       // Create a Handle
       Cup.Handle h = /* Your code goes here */ ;
   }
}
```

Solution:

```
// CupTest.java
package com.jdojo.innerclasses.exercises;

public class CupTest {
   public static void main(String[] args) {
       // Create a Cup
       Cup c = new Cup();

       // Create a Handle
       Cup.Handle h = c.new Handle();
   }
}
```

4. What will be the output when the following `Outer` class is run?

```
// Outer.java
package com.jdojo.innerclasses.exercises;

public class Outer {
    private final int value = 19680112;

    public class Inner {
        private final int value = 19690919;
            public void print() {
            System.out.println("Inner: value = " + value);
            System.out.println("Inner: this.value = " + this.value);
            System.out.println("Inner: Inner.this.value = " +
                                Inner.this.value);
            System.out.println("Inner: Outer.this.value = " +
                                Outer.this.value);
        }
    }

    public void print() {
        System.out.println("Outer: value = " + value);
        System.out.println("Outer: this.value = " + this.value);
        System.out.println("Outer: Outer.this.value = " +
                           Outer.this.value);
    }

    public static void main(String[] args) {
        Outer out = new Outer();
        Inner in = out.new Inner();
        out.print();
        in.print();
    }
}
```

Answer:

```
Outer: value = 19680112
Outer: this.value = 19680112
Outer: Outer.this.value = 19680112
Inner: value = 19690919
Inner: this.value = 19690919
Inner: Inner.this.value = 19690919
Inner: Outer.this.value = 1968011
```

5. The following declaration of an `AnonymousTest` class does not compile. Describe the reasons and steps you might take to fix the error.

```
// AnonymousTest.java
package com.jdojo.innerclasses.exercises;

public class AnonymousTest {
   public static void main(String[] args) {
       int x = 100;

       Object obj = new Object() {
           {
               System.out.println("Inside. x = " + x);
           }
       };

       x = 300;
       System.out.println("Outside. x = " + x);
   }
}
```

Answer:

Compilation fails because the anonymous class is accessing local variable `x`, which is not effectively final. To fix the error, make `x` effectively final by removing assignment x = 300;

6. Consider the following declaration for a top-level class `A` and a member inner class `B`:

```
// A.java
package com.jdojo.innerclasses.exercises;

public class A {
    public class B {
        public B() {
            System.out.println("B is created.");
        }
    }

    public A() {
        System.out.println("A is created.");
    }
}
```

Consider the following incomplete declaration of class `C`, which inherits from the inner class `A.B`:

```
// C.java
package com.jdojo.innerclasses.exercises;

public class C extends A.B {

   /* Define a constructor for class C here */

   public static void main(String[] args) {
       C c = /* Your code goes here */;
   }
}
```

Add an appropriate constructor for class `C` and complete the statement in the `main()` method. When class `C` is run, it should print the following to the standard output:

```
A is created.
B is created.
C is created.
```

Solution:

```
// C.java
package com.jdojo.innerclasses.exercises;

public class C extends A.B {
    public C(A a) {
        a.super();
        System.out.println("C is created");
    }

    public static void main(String[] args) {
       C c = new C(new A());
    }

 }
```

7. Which of the following is true about an anonymous inner class?

   a. It can inherit from one class and implement one interface.

   b. It can inherit from one class and implement multiple interfaces.

   c. It can inherit from one class or implement one interface.

   d. It can implement multiple interfaces, but inherits from only one class.

Answer:

The option `c` is correct.

8. How many class files will be generated when the following declaration of the `Computer` class is compiled? List the names of all generated class files.

```
// Computer.java
package com.jdojo.innerclasses.exercises;

public class Computer {
    public class Mouse {
        public class Button {
        }
    }

   public static void main(String[] args) {
       Object obj = new Object() {
       };

       System.out.println(obj.hashCode());
   }
}
```

Answer:

Four class fileswill be generated

- `Computer.class`
- `Computer$Mouse.class`
- `Computer$Mouse$Button.class`
- `Computer$1.class`

9. The following declaration of class H does not compile. Point out the problem and suggest a solution.

```
// H.java
package com.jdojo.innerclasses.exercises;

public class H {
   private int x = 100;

   public static class J {
       private int y = x * 2;
   }
}
```

Answer:

Compilation fails because static inner class J is accessing non-static variable x of outer class H. To fix error, make x static OR make class J non-static.

10. Consider the following declaration of a top-level class P and a nested static class Q:

```
// P.java
package com.jdojo.innerclasses.exercises;

public class P {
   public static class Q {
       {
            System.out.println("Hello from Q.");
       }
   }
}
```

Complete the main() method of the following PTest class that will create an object of the nested static class Q. When class PTest is run, it should print a message "Hello from Q." to the standard output.

```
// PTest.java
package com.jdojo.innerclasses.exercises;

public class PTest {
   public static void main(String[] args) {
       P.Q q = /* Your code goes here */;
   }
}
```

Solution:

```
// PTest.java
package com.jdojo.innerclasses.exercises;

public class PTest {
   public static void main(String[] args) {
       P.Q q = new P.Q();

  }
}
```