PGCert IT: Programming for Industry

Exception Handling Answers - Exercise One and Four(5)

Exercise One: Try & Catch

1. What is the problem with the following code?

```
private void tryCatch01() {
  int result = 0;
  int[] nums = null;
  try {
    result = nums.length;
    System.out.println("See you");
  } catch (ArithmeticException e) {
    System.out.println("Problem");
    result = -1;
  }
  System.out.println("Result: " + result);
}
```

Catching `ArithmeticException`, which is the wrong type of exception (`NullPointerException` will be thrown).

2. Rewrite the following code, adding an appropriate try-catch block to it:

```
private void tryCatch02() {
  int num1 = 120, num2 = 120, result = 0;
  result = num2 / (num1 - num2);
  System.out.println("Result: " + result);
}
```

```
private void tryCatch02() {
      int num1 = 120, num2 = 120, result = 0;
      try {
         result = num2 / (num1 - num2);
      } catch (ArithmeticException e) {
         // Handle the error...
      }
      System.out.println("Result: " + result);
}
```

3. Rewrite the following code, adding an appropriate try-catch block to it:

```
private void tryCatch03() {
  int result = 0;
  String[] items = { "one", "two", null };
  result = items[2].length();
  System.out.println("Result: " + result);
}
```

```
private void tryCatch03() {

      int result = 0;
      String[] items = { "one", "two", null };
      try {
         result = items[2].length();
      } catch(NullPointerException e) {
         // Handle the error...
      }
      System.out.println("Result: " + result);
}
```

4. Correct the errors in the following code:

```
private void tryCatch04() {
  int num = 0; // Moved from within the try block
  try {
    System.out.println("Enter number: ");
    num = Integer.parseInt(Keyboard.readInput());
    System.out.println("Thank you");
  } catch (NumberFormatException e) {
    System.out.println("Input error");
    num = -1;
  }
  System.out.println("Number: " + num);
}
```

5. Correct the errors in the following code:

```
private int tryCatch05() {
  int result = 0;
  int[] items = new int[]{ 2, 3, 4, -1, 4 };
  try {
    result = nums[nums[3]];
    System.out.println("See you");
  } catch (ArrayIndexOutOfBoundsException e) {
    System.out.println("Number error");
    result = -1;
  }
  return result;
}
```

6. What is the output of the following code, when `tryCatch06()` is called?

```
private void tryCatch06() {
  try {
    try06(0, "");
    System.out.println("A");
  } catch (ArithmeticException e) {
    System.out.println("B Error");
  }
}

private void try06(int num, String s) {
  System.out.println("C");
  try {
    num = s.length();
    num = 200 / num;
  } catch (NullPointerException e) {
    System.out.println("E Error");
  }
    System.out.println("F");
}
```

```
C
B Error
```

7. In the code below, where should you put the try-catch if you *always* want the statement `System.out.println("C")` to be executed, even if there is an exception in the statement `num = s.length()` ?

```
private void tryCatch07() {

  try07(0, null);

  System.out.println("A");

}

private void try07(int num, String s) {
  System.out.println("B");

  num = s.length();

  System.out.println("C");
}
```

Around the line num = s.length();

8. What is the output of the following code, when `tryCatch08()` is called?

```
private void tryCatch08() {
  try {
    try08(0, null);
    System.out.println("A");
  } catch (NullPointerException e) {
    System.out.println("B");
  }
}

private void try08(int num, String s) {
  System.out.println("C");
  try {
    num = s.length();
    System.out.println("D");
  } finally {
    System.out.println("E");
  }
  System.out.println("F");
}
```

```
C
E
B
```

9. What is the output of the following code, when `throwsClause09()` is called?

```
private void throwsClause09() {
  try {
    throws09(null);
    System.out.println("A");
  } catch (NullPointerException e) {
    System.out.println(e);
  }
  System.out.println("B");
}

private void throws09(String numS) throws NullPointerException {
  if (numS == null) {
    throw new NullPointerException("Null String");
  }
  System.out.println("C");
}
```

```
java.lang.NullPointerException: Null String
B
```

10. What is the output of the following code, when `throwsClause10()` is called?

```
private void throwsClause10() {
  try {
    throws10(null);
    System.out.println("A");
  } catch (ArithmeticException e) {
    System.out.println(e);
  } finally {
    System.out.println("B");
  }
  System.out.println("C");
}

private void throws10(String numS) throws NullPointerException {
  if (numS == null) {
    throw new NullPointerException("Bad String");
  }
  System.out.println("D");
}
```

```
B

And then the program will crash because of the unhandled NullPointerException.
When the program crashes, it will print “java.lang.NullPointerException: Bad
String”, followed by a stack trace.
```

Exercise Four: Simple Exceptions

5. What is the output of the program on the following page? Explain *why* this is the output.

```
public class SimpleExceptions2 {

	public static void main(String[] args) {
		SimpleExceptions2 exceptions = new SimpleExceptions2();
		exceptions.question2();
	}

	public void question2() {
		try {
			System.out.print("1: ");
			perform("3");
			System.out.print("A ");
			System.out.println();
```

```
                System.out.print("2: ");
                perform("0");
                System.out.print("B ");
                System.out.println();

                System.out.print("3: ");
                perform(null);
                System.out.print("C ");
                System.out.println();

                System.out.print("4: ");
                perform("");
                System.out.print("D ");
                System.out.println();
            } catch (NullPointerException e) {
                System.out.print("E ");
            } catch (Exception e) {
                System.out.print("F ");
            }
        }

        private void perform(String input) {
            try {
                int length = input.length();
                int num1 = Integer.parseInt(input);
                System.out.print("A4 ");
                int num2 = 100 / num1;
                System.out.print("B4 ");
            } catch (NumberFormatException e) {
                System.out.print("C4 ");
            } catch (ArithmeticException e) {
                System.out.print("D4 ");
            } finally {
                System.out.print("E4 ");
            }
            System.out.print("F4 ");
        }
}
```

```
1: A4 B4 E4 F4 A

2: A4 D4 E4 F4 B

3: E4 E
```

The first method call has no exception; The second method call throws ArithmeticException, which is handled by the try and catch block inside the perform method. The third method call throws NullPointerException, where the perform method cannot handle and the exception is parsed to the next point of call. The NullPointerException is then handled by the try and catch block in the question2 method.