

COMP 250

INTRODUCTION TO COMPUTER SCIENCE

Assignment Project Exam Help

<https://powcoder.com>

Week 3-1: Errors, exceptions, and try catch blocks

Add WhatsApp powder

Giulia Alberini, Fall 2020

WHAT ARE WE GOING TO DO IN THIS VIDEO?



- Errors and Exceptions

Assignment Project Exam Help

- Try/catch blocks

<https://powcoder.com>

Add WeChat powcoder

KIND OF ERRORS

- **Stylistic Errors**

The functionality of your code is not affected.

Your code is hard to read.

- **Compile-time errors**

Something is wrong with the syntax of the program.

Your code does not compile!

- **Run-time errors**

Something goes wrong when you run your program.

Your code does not run!

- **Logic errors**

Something isn't working the way you think.

The program does not do the correct thing.

COMPILE TIME ERRORS

- These errors are detected by the compiler.

Assignment Project Exam Help

- They prevent your code from running since the program violates the syntactic rules of Java.

<https://powcoder.com>

Add WeChat powcoder

- Compile-time errors include missing a semicolon, adding an extra bracket, or using the incorrect types.
- Compile-time errors are the easiest to fix!

RUN-TIME ERRORS

- These errors happen at run time. They are detected by the JVM when it tries to execute the instructions in the program.
- Common run-time errors include index out of bounds exceptions, division by zero, or null pointer exceptions.
- They are more difficult to fix than compile-time errors, but the JVM output some useful text to help you with that.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

EXCEPTIONS

Java displays a message with:

- the name of the exception,
- the line of the program where it occurred, and
- a "stack trace": the list of the method calls the application was in the middle of when an Exception was thrown.

```
java.lang.StringIndexOutOfBoundsException: String index out of range: 3
    at java.lang.String.charAt(Unknown Source)
    at Test.method3(Test.java:19)
    at Test.method2(Test.java:13)
    at Test.method1(Test.java:9)
    at Test.main(Test.java:4)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

Assignment Project Exam Help

THROWING EXCEPTIONS

<https://powcoder.com>

Add WeChat powcoder

RUN TIME ERRORS AND EXCEPTIONS

An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Exceptions happen when the computer is asked to do something that doesn't make sense.

AN EXAMPLE

Assignment Project Exam Help

```
int[] x = {1, 2, 3};  
System.out.println(x[3]);
```

<https://powcoder.com>
Add WeChat powcoder

On the second line we try to access an element that does not exists. An `ArrayIndexOutOfBoundsException` will be thrown.

THROWING EXCEPTIONS

Assignment Project Exam Help

When you want to communicate to a user that an invalid input was passed, or that some error occurred during the execution of the program, you can decide to deliberately stop the execution of your code by throwing an exception.

<https://powcoder.com>

Add WeChat powcoder

EXAMPLE

```
public static boolean isPrime (int n) {  
    if (n <= 1) {  
        return false;  
    }  
    ...  
}
```

Do we want to return false? We can instead throw an exception!

EXAMPLE

```
public static boolean isPrime (int n) {  
    if (n <= 1) {  
        throw new IllegalArgumentException("Primality  
is not defined for number smaller than 2");  
    }  
    ...  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

EXAMPLE

```
public static boolean isPrime (int n) {  
    if (n <= 1) {  
        throw new IllegalArgumentException("Primality  
is not defined for number smaller than 2");  
    }  
    ...  
}
```

Keywords

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

EXAMPLE

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

```
public static boolean isPrime (int n) {  
    if (n <= 1) {  
        throw new IllegalArgumentException("Primality  
is not defined for number smaller than 2");  
    }  
    ...  
}
```

Keywords

Exception name

EXAMPLE

```
public static boolean isPrime (int n) {  
    if (n <= 1) {  
        throw new IllegalArgumentException("Primality  
is not defined for a number smaller than 2");  
    }  
    ...  
}
```

Message that will be displayed

THROWING EXCEPTIONS

You can throw any kind of Exception you want.

Assignment Project Exam Help

Example:

<https://powcoder.com>
ArithmeticException, ArrayIndexOutOfBoundsException,
ArrayStoreException, ClassCastException, IllegalArgumentException,
IllegalMonitorStateException, IllegalStateException,
IllegalThreadStateException, IndexOutOfBoundsException,
NegativeArraySizeException, NullPointerException,
NumberFormatException, ...

Add WeChat powcoder

THROWING EXCEPTIONS

In general, if you want to throw an exception you can add the following statement in your code:

<https://powcoder.com>

```
throw new IllegalArgumentException(message)
```

The type of message is `String` and the message is optional.

Assignment Project Exam Help
TRY/CATCH BLOCKS

<https://powcoder.com>

Add WeChat powcoder

DEALING WITH EXCEPTIONS

What can we do to prevent our code from crashing at run-time? **Assignment Project Exam Help**

<https://powcoder.com>
Add WeChat powcoder

```
public static void myMethod(int[] x) {  
    if (x != null) {  
        // something with x  
    }  
}
```

What type of exception are we trying to prevent?

Is there another way?

CATCHING EXCEPTIONS

In order to prevent our code from crashing, we can *catch* Exceptions as follows:

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

e is the name I gave
to my variable of
type Exception

```
try {  
    // code that might be problematic  
}  
catch(Exception e) {  
    // code that should be executed if there was a problem  
}  
// whatever comes after
```

TRY/CATCH BLOCKS

Assignment Project Exam Help

A try/catch block allows us to try some code, and if an exception is raised, we can then "catch" the exception.

<https://powcoder.com>

An exception that is caught will not cause the program to crash.

Add WeChat powcoder

EXAMPLE 1

Assignment Project Exam Help

```
int[] x = {1,2,3};  
try {  
    System.out.println(x[3]);  
}  
catch (ArrayIndexOutOfBoundsException e) {  
    System.out.println("Wrong index!");  
}  
System.out.println("Everything else");
```

<https://powcoder.com>

Add WeChat powcoder

What prints?

➤ Wrong index!
Everything else

EXAMPLE 2

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder ➤ Run-time error!

```
int[] x = {1,2,3};  
try {  
    System.out.println(x[3]);  
}  
catch (NullPointerException e) {  
    System.out.println("It is null!");  
}  
System.out.println("Everything else");
```

What prints?

`ArrayIndexOutOfBoundsException`
is thrown and it is not caught
by any catch block.

CATCHING MULTIPLE EXCEPTIONS

If we want to do something different depending on the type of the exception caught, we can use multiple catch blocks.

Assignment Project Exam Help

```
try {  
    // code that might be problematic  
}  
catch (ArrayIndexOutOfBoundsException e) {  
    // what to do in this case  
}  
catch (NullPointerException e) {  
    // what to do in this other case  
}  
// whatever comes after
```

<https://powcoder.com>

Add WeChat powcoder

EXAMPLE

```
int[] x = {1,2,3};
try {
    System.out.println(x[1]/0);
}
catch (ArrayIndexOutOfBoundsException e) {
    System.out.println("Wrong index!");
}
catch (ArithmeticException e) {
    System.out.println("Bad math");
}
System.out.println("Everything else");
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

What prints?

➤ Bad math

Everything else

CATCHING ALL EXCEPTIONS

If we would like to catch all possible exceptions with the same catch block we can do the following:

Assignment Project Exam Help

```
try { https://powcoder.com  
    // code that might be problematic  
}  
    Add WeChat powcoder  
catch(Exception e) {  
    // what to do in case of issues  
}  
// whatever comes after
```

All different exceptions are considered to be of type `Exception`.

WHAT ABOUT ALL THE INFO?

NOTE: `e` is the name of the variable name used in the catch block.

- If you would like to display all the information related to some caught exception (what you would normally see in red when the code crashes), you can use the following method:

Assignment Project Exam Help

```
https://powcoder.com  
e.printStackTrace();
```

Add WeChat powcoder

- If you want to just print the name of the exception you can use a `print/println` statement:

```
System.out.println(e);
```

finally BLOCK

- The finally block **always** executes when the try block exists. This ensures that the finally block is executed even if one of the following happens:
 - an unexpected exception occurs in the try block
 - an exception occurs in the catch block
 - There's a return/continue/break statement in the try/catch block.
- You can have a finally even with just a try block (and no catch).
- The finally block is useful for more than just exception handling.
Good practice: put cleanup code in a finally block even if no exception is anticipated.

CHECKED
Assignment Project Exam Help
VS

<https://powcoder.com>

EXCEPTIONS
Add WeChat powcoder

CHECKED VS UNCHECKED EXCEPTIONS

In java there are two kinds of exceptions:

- **Checked**

- Exception
- IOException
- FileNotFoundException

- **Unchecked**

- NullPointerException
- ArrayIndexOutOfBoundsException
- ArithmeticException

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

UNCHECKED EXCEPTIONS

- These exceptions are **not** checked at compile-time.
Assignment Project Exam Help
- Most (if not all) exceptions we have seen up to now are unchecked, they can cause your code to crash at run-time.
**<https://powcoder.com>
Add WeChat powcoder**
- You are not forced by the compiler to handle these exceptions. It is up to the programmer to decide if to catch the exceptions.

CHECKED EXCEPTIONS

- These exceptions are checked at compile-time!

Assignment Project Exam Help

- Usually, these kinds of Exceptions happen due to something that isn't the programmer's fault. For example, trying to read from a file whose name is misspelled. These are considered 'recoverable' errors, which is why you have to handle them - it might be possible to keep going and not just crash!

<https://powcoder.com>

Add WeChat powcoder

CHECKED EXCEPTIONS

The programmer is forced to *handle* these exceptions. There are 2 ways to do that:

Assignment Project Exam Help

1. Use try/catch block to surround the code that might throw a checked exception

<https://powcoder.com>
Add WeChat powcoder

2. Specify in the method header that the method contains code that might throw an exception, and therefore the method itself (might) **throws** an exception.

EXAMPLE – READING FROM A FILE

You can read from a file in Java using the same `Scanner` class that we use to read from standard input!

You use an overloaded constructor inside the `Scanner` class.

`File` is an object meant to store information related to the `File`. You can create one by providing a path to a file.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Scanner

```
public Scanner(File source)
    throws FileNotFoundException
```

Constructs a new `Scanner` that produces values scanned from the specified file.

Parameters:

`source` - A file to be scanned

Throws:

`FileNotFoundException` - if `source` is not found

EXAMPLE – READING FROM A FILE

Notice the "throws" keyword in the method header. This means that you need to *handle* a `FileNotFoundException` to avoid the compiler error.

Assignment Project Exam Help

Scanner <https://powcoder.com>

```
public Scanner(File source)  
    throws FileNotFoundException
```

Constructs a new Scanner that produces values scanned from the specified file.

Parameters:

source - A file to be scanned

Throws:

`FileNotFoundException` - if source is not found

EXAMPLE

For example,
filePath **could be**
"C:\\documents\\
\\whatever.txt"

We would like to write the method below, but we get the following compile-time error:

Assignment Project Exam Help
<https://powcoder.com>
unreported exception java.io.FileNotFoundException;
must be caught or declared to be thrown

Add WeChat powcoder

```
public static void myMethod(String filePath) {  
    File f = new File(filePath);  
    Scanner fileReader = new Scanner(f);  
    int firstNumber = fileReader.nextInt();  
}
```

OPTION 1

Surround the code that might throw an exception with a try/catch block.

Assignment Project Exam Help

```
public static void myMethod(int x) {  
    try {  
        File f = new File(filePath);  
        Scanner fileReader = new Scanner(f);  
        int firstNumber = fileReader.nextInt();  
    }  
    catch (FileNotFoundException e) {  
        System.out.println("File not found");  
    }  
}
```

<https://powcoder.com>

Add WeChat powcoder

OPTION 2

Specify in the method header that there's an exception using the `throws` keyword followed by the type of the exception.

Assignment Project Exam Help

```
public static void myMethod(String filePath) throws FileNotFoundException {  
    File f = new File(filePath);  
    Scanner fileReader = new Scanner(f);  
    int firstNumber = fileReader.nextInt();  
}
```

Whenever this method is called, the `FileNotFoundException` will have to be handled. To do so, you can use any of the two options we have just seen.

EXAMPLE

```
public static void test() throws FileNotFoundException {  
    // code from the previous slide  
}  
  
public static void test2() throws FileNotFoundException {  
    :  
    test();  
    :  
}  
  
public static void test3() throws FileNotFoundException {  
    :  
    test2();  
    :  
}  
  
public static void main(String[] args) throws FileNotFoundException {  
    test3();  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
public static void test() throws FileNotFoundException {  
    // code from the previous slide  
}
```

```
public static void test2() throws FileNotFoundException {  
    :  
    test();  
    :  
}
```

```
public static void test3() throws FileNotFoundException {  
    :  
    test2();  
    :  
}
```

```
public static void main(String[] args) {  
    try{  
        test3();  
    } catch(FileNotFoundException e) {  
        System.out.println("Caught here!");  
    }  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder


```
public static void test() throws FileNotFoundException {  
    // code from the previous slide  
}  
  
public static void test2() throws FileNotFoundException {  
    :  
    test();  
    :  
}  
  
public static void test3() {  
    :  
    try{  
        test2();  
    } catch(FileNotFoundException e) {  
        System.out.println("Caught here!");  
    }  
    :  
}  
  
public static void main(String[] args) {  
    test3();  
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Coming Soon

Assignment Project Exam Help

In the next video we will start to talk about objects and classes.

<https://powcoder.com>

Add WeChat powcoder