

COMP 250

Assignment Project Exam Help

INTRODUCTORY SCIENCE

<https://eduassistpro.github.io/>

Week 3-1: Errors, exceptions, ch blocks

Giulia Alberini, Fall 2020

WHAT ARE WE GOING TO DO IN THIS VIDEO?



- Errors and Exceptions
- Try/catch blocks

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

KIND OF ERRORS

- **Stylistic Errors**

The functionality of your code is not affected.

Your code is hard to read.

- **Compile-time**

Something is wrong with your

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.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

COMPILE TIME ERRORS

- These errors are detected by the compiler.

Assignment Project Exam Help

- They prevent your the program violates the syntactic rules o <https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

- 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 array 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 outputs some useful text to help you with that.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

EXCEPTIONS

Java displays a message with:

- the name of the exception,
- the line of the program and
- a "stack trace": the list of application was in the middle of when an exception occurred

```
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

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

RUN TIME ERRORS AND EXCEPTIONS

An exception is an event, which occurs during the execution of a program that interrupts the normal flow of the program.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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

AN EXAMPLE

Assignment Project Exam Help

```
int[  
System.out.println(  
    https://eduassistpro.github.io/  
    Add WeChat edu_assist_pro
```

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 check whether an invalid input was passed during the execution of the program, you can deliberately stop the execution of your code by throwing an exception.

<https://eduassistpro.github.io/>

Add WeChat [edu_assist_pro](#)

EXAMPLE

```
public static boolean isPrime (int n) {  
    if (n <  
        retu https://eduassistpro.github.io/  
    }  
    ...    Add WeChat edu_assist_pro  
}
```

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://eduassistpro.github.io/>

Add WeChat edu_assist_pro

EXAMPLE

```
public static boolean isPrime (int n) {  
    if (n <= 1) {  
        throw new I tion("Primality  
is not defined for number sn an 2");  
    }  
    ...  
}
```

Keywords

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

EXAMPLE

Assignment Project Exam Help
<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

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

Keywords

Exception name

EXAMPLE

Assignment Project Exam Help
<https://eduassistpro.github.io/>
Primalty
is not defined for a number less than 2
Add WeChat edu_assist_pro

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

Message that will be displayed

THROWING EXCEPTIONS

You can throw any kind of Exception you want.

Assignment Project Exam Help

Example:

ArithmeticException, **ArrayStoreException**, **ClassCastException**, **IllegalArgumentException**,
IllegalMonitorStateException, **IllegalThreadStateException**, **IndexOutOfBoundsException**,
NegativeArraySizeException, **NullPointerException**,
NumberFormatException, ...

<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

THROWING EXCEPTIONS

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

<https://eduassistpro.github.io/>

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

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

Assignment Project Exam Help
TR <https://eduassistpro.github.io/> CKS
Add WeChat edu_assist_pro

DEALING WITH EXCEPTIONS

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

```
public static void main(String[] args) {  
    if (args != null) {  
        // something with args  
    }  
}
```

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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, the exception.

An exception that is caught will prevent the program to crash.

EXAMPLE 1

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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

What prints?

➤ Wrong index!

Everything else

EXAMPLE 2

Assignment Project Exam Help

<https://eduassistpro.github.io/> What prints?

Add WeChat edu_assist_pro Non-time error!

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

`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 t  
}  
catch (ArrayIndexOutOfBoundsException e) {  
    // what to do in this case  
}  
catch (NullPointerException e) {  
    // what to do in this other case  
}  
// whatever comes after
```

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

EXAMPLE

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

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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 {  
    // c  
} catch(Exception e) {  
    // what to do in case of issues  
}  
// whatever comes after
```

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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://eduassistpro.github.io/>;

Add WeChat edu_assist_pro

- 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
 - an exception occurs
 - There's a **return/continue/break** 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://eduassistpro.github.io/>

EXCEPT
Add WeChat edu_assist_pro

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://eduassistpro.github.io/>

Add WeChat edu_assist_pro

UNCHECKED EXCEPTIONS

- These exceptions are **not** checked at compile-time.

Assignment Project Exam Help

- Most (if not all) exceptions that are unchecked, they can cause your code to crash at run-time.

Add WeChat edu_assist_pro

- 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 kind of errors are due to something that isn't the programme trying to read from a file whose name is misspelled. These are 'recoverable' errors, which is why you have to handle them - it might be possible to keep going and not just crash!

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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 that might throw a checked exception

<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

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!

Assignment Project Exam Help

<https://eduassistpro.github.io/>

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

Add WeChat edu_assist_pro

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

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

<https://eduassistpro.github.io/>

Add WeChat `edu_assist_pro`

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:

unreported exception FileNotFoundException;
must be thrown

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

```
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 main(String[] args) {  
    try {  
        File f = new File("data.txt");  
        Scanner fileReader = new Scanner(f);  
        int firstNumber = fileReader.nextInt();  
    }  
    catch (FileNotFoundException e) {  
        System.out.println("File not found");  
    }  
}
```

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

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() 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://eduassistpro.github.io/>

Add WeChat edu_assist_pro

```
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://eduassistpro.github.io/>

Add WeChat edu_assist_pro


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

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

An orange paint roller with a red handle, positioned horizontally. The roller is covered in orange paint, which is dripping down the left side. The words "Coming Soon" are written in white, sans-serif font across the middle of the roller.

Coming Soon

Assignment Project Exam Help

In the next video

t objects and classes.

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro