How to Handle Exceptions

Step 1

Place a try block around the statement(s) that may throw the exception.

try { ... }

Step 2

Place a catch block right after the try block.

```
catch (... Exception e) {
...
}
```

Throwing Exceptions

Question

How can the client throw an exception?

900

Question

How can the client throw an exception?

Answer

throw new ...Exception(...);

Question

How can the client throw an exception?

Answer

throw new ...Exception(...);

Question

Why would a client ever throw an exception?

Question

How can the client throw an exception?

Answer

throw new ...Exception(...);

Question

Why would a client ever throw an exception?

Answer

For example, the client may want to separate the error handling code from the rest.

Write a program that reads from the user a mathematical expression that may contain nested precedence characters, which are the following: parentheses (and), brackets [and], and braces { and }.

For example, the input string can be

 $a-b*[c/(d+e)]+f*[g-h*{i+j*k}]-m/(n+1)/w$

The program should ignore all other characters and focus only on the above precedence characters. Specifically, it should determine whether these characters are "OK", "Imbalanced", "Overlapping" or "Too Deeply Nested".

Check11D

Enter expression [a+b*(a-d]) Overlapping

Enter expression (a+(b-[a/d]) Imbalanced

Enter expression (a+b)]-a Imbalanced

```
Enter expression
{a-[b/(a+d)]}
OK
```

```
Enter expression
(((([[a+b]/a]+d)+e)+f)
Too Deeply Nested
```

Check11D

- Create an instance of the type.lib.CharStack class, an ordered collection of chars, using its default constructor.
- Scan the input string character by character, left to right.
- Ignore characters different from the precedence characters.
- If an open precedence character is encountered (i.e., (or [or {), store something on the stack.
- If a close precedence character is encountered, remove from the stack the character that was stored last. If the removed character corresponds to the close precedence character then continue scanning. Otherwise, end the program with the "Overlapping" message.

- If you need to remove a character from the stack but find it empty, or you complete scanning the string but find the stack not empty, then end the program with the "Imbalanced" message.
- If you need to store something in the stack but find it full, then end the program with the "Too Deeply Nested" message.
- If the program did not end in one of the above cases, print "OK."

A Multiclass Application CSE 1020

moodle.yorku.ca

moodle.yorku.ca CSE 1020

Generate UML class diagrams for classes from their code.

Generate UML class diagrams for classes from their code.

Problem

Generate a UML class diagram for a class and its components (aggregation) from its code.

Definition

Aggregation is a binary relation on classes. The pair (A, P) of classes is in the aggregation relation if class A (aggregate) has an attribute of type P (part).

The aggregation relation is also known as the has-a relation. Instead of saying that (A, P) is in the aggregation relation, we often simply say that A has-a P.

Prompt the user "Enter class name: ", read the user's input, and print the names of all components of the class. Reprompt the user if the class cannot be found.

Prompt the user "Enter class name: ", read the user's input, and print the class and all its components using <u>PlantUML</u>.

©startuml class AbstractFoods interface List interface Map AbstractFoods o-- Map AbstractFoods o-- List AbstractFoods o-- Map @enduml

▶ ∢ ∃ ▶



<ロ> (四) (四) (三) (三)

2

```
StringBuffer diagram = ...;
SourceStringReader reader =
    new SourceStringReader(diagram.toString());
String fileName = ...;
File file = new File(fileName);
OutputStream stream = new FileOutputStream(file);
reader.generateImage(stream);
```

• Study the remainder of Chapter 11.