CSE1720

Week 01, Lecture 01

Winter 2014 ◆ Tuesday, Jan 6, 2014

YORK UNIVERSITÉ UNIVERSITY

Objectives for this class meeting

1. Review and discussion of course syllabus
2. Presentation of course roadmap
   - Discussion and feedback
3. Topic: Exceptions

YORK UNIVERSITÉ UNIVERSITY

CSE 1720 3.0 Building Interactive Systems

This course continues an introduction to computer programming within the context of image, sound and interaction, subsequent to CSE1710 3.0. The student's foundation in basic programming will serve as a platform from which to explore the use of diverse media within interactive systems, including the WWW and simple game systems.

Topics include:

- User Interfaces (UIs)
- UI Elements
- Event driven programming
- Intro to threads
- User Interface Builders
- Guidelines for UI design
- Objects, classes and inheritance
- Interactive WWW-based systems - introduction to WWW and basic network concepts, HTML, Javascript, other WWW technologies (e.g. Flash), guidelines for WWW design
- How to design simple games and make them engaging

Prerequisites: CSE1710 3.0 *Course Credit Exclusions:* CSE1020 3.0, ITEC1620 3.0, ITEC1630 3.0

YORK UNIVERSITÉ UNIVERSITY

3

Course Roadmap

- Exceptions (Ch 11)
- Collections (Ch 8)
- Graphics2D services (Ch 8)
- Event Driven Programming, Graphical Animations, the Observer Design Pattern
- Inheritance, Polymorphism (Ch 9)
- The Collection Framework (Ch 10)
- Data Models (for Game Design), decoupling the model from the view
- Interactive Game Design, Model-View-Control Architecture
- Principles of Game Design
- Introducing widgets, Principles of UI design

YORK UNIVERSITÉ UNIVERSITY

4

Topic

- exception handling – Chapter 11

What is a *clean exit*?
What is a *crash*?

- A clean exit is when an app terminates in a controlled and orderly manner
  - *flow of control* has reached the end of the main method; there are no more bytecode instructions to execute, so the VM begins **shutdown**.
    - flush all output buffers
    - complete all pending transactions
    - close all network connections
    - free up all used resources
- A *crash* is a non-clean exit
  - abrupt termination; flow of control has not reached the end of the bytecode
  - may be accompanied by error messages (e.g., the stack trace)
  - a crash occurs due to an **unhandled exception**

What Are Exceptions? (I)

- An exception is a *situation* of an **error state** that has arisen to the VM during runtime.
- An exception is encapsulated by an object
- The object gets instantiated in these error state situations; the object's state captures information about the error state
- One example of an **error state** is attempting to perform an illegal operation, such as integer/long quotient by zero
  - also possibly integer/long modulo operation
  - **NOT** double/float point division or modulo (these operations have *closure*)

Example: The Quotient app

- Here is a fragment of an app that prompts for two integers, and then computes and output their quotient.

```
output.println("Enter the first integer:");
int a = input.nextInt();
output.println("Enter the second:");
int b = input.nextInt();

int c = a / b;
output.println("Their quotient is: " + c);
```

- If the user inputs 0 for the second integer, observe the outcome

Other Error States…

- Another example of an **error state** is when the Virtual Machine itself runs into a problem
  - for instance, the VM runs out of run-time memory

YORK UNIVERSITÉ UNIVERSITY

Other Error States…

- Another example of an **error state** is not catching an exception that is thrown by a service

substring

```
public String substring(int beginIndex)
```

Returns a new string that is a substring of this string. The substring begins with the character at the specified index and extends to the end of this string.

Examples:

```
"unhappy".substring(2) returns "happy"
"Harbison".substring(3) returns "bison"
"emptiness".substring(9) returns "" (an empty string)
```

Parameters:
`beginIndex` - the beginning index, inclusive.
Returns:
the specified substring.
Throws:
`IndexOutOfBoundsException` - if `beginIndex` is negative or larger than the length of this `String` object.