

CSSE 220 Day 22

Exam 2 Review
File I/O, Exceptions
Vector Graphics Project

Check out *FilesAndExceptions* from SVN

Questions?

Today

- ▶ Exam 2 review
 - ▶ File I/O and Exceptions
 - ▶ Vector Graphics project kickoff
- 

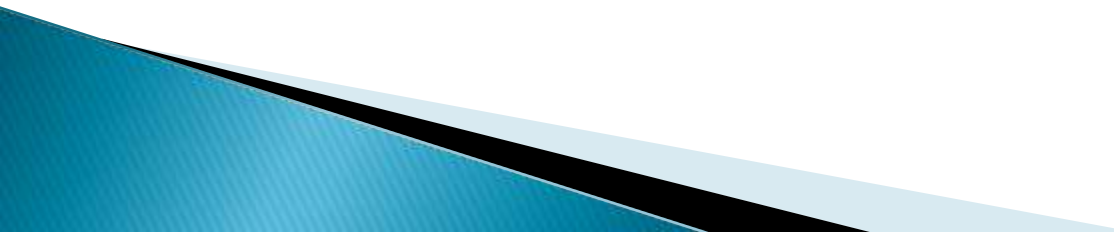
Exam 2 Review

»» Exam is in class tomorrow ...

Exam Coming!

- ▶ Test on Tuesday
 - In-class exam
 - Meet in your usual room at time indicated on the schedule
- ▶ Topics from Chapters 1–14
- ▶ Will include:
 - An open computer exam with:
 - A paper part: short answer, fill-in-the-blank, trace-code-by-hand, draw box-and-pointer diagrams, find-errors-in-code, write short chunks of code, etc
 - A programming part: a few small programs, unit tests provided for some of them, you write unit tests for others
- ▶ Review in class today
 - What questions did you bring?
 - What topics would you like to review?
 - I didn't prepare anything but I'm happy to cover whatever you want, including working examples

Have you done these?

- ▶ Reviewed chapters 1 to 14 from Big Java
 - ▶ Prepared a sheet of notes to help you summarize what you consider important
 - ▶ Reviewed the slides, in-class quizzes, homework from sessions 1 to 21
 - ▶ Practiced programming, unit testing, documenting your code, & using the Java API
 - ▶ You can ask questions by email to the csse220-staff mailing list or your instructor
- 

Files and Exceptions

- » Reading & writing files
- When the unexpected happens

Review of Anonymous Classes

- ▶ Look at GameOfLifeWithIO
 - GameOfLife constructor has 2 listeners, one *local inner* class and one *local anonymous* class
 - ButtonPanel constructor has 3 listeners which are *local anonymous* classes
- ▶ Feel free to use as examples for your project

File I/O: Key Pieces

- ▶ Input: **File** and **Scanner**
- ▶ Output: **PrintWriter** and **println**
- ▶ Be kind to your OS: **close()** all files
- ▶ Letting users choose: **JFileChooser** and **File**
- ▶ Expect the unexpected: **Exception** handling
- ▶ Refer to examples when you need to...

Exceptions

- ▶ Used to signal that something went wrong:
 - `throw new EOFException("Missing column");`
- ▶ Can be **caught** by **exception handler**
 - Recovers from error
 - Or exits gracefully

A Checkered Past

- ▶ Java has two sorts of exceptions
- ▶ **Checked exceptions**: compiler checks that calling code isn't ignoring the problem
 - Used for **expected** problems
- ▶ **Unchecked exceptions**: compiler lets us ignore these if we want
 - Used for **fatal** or **avoidable** problems
 - Are subclasses of RuntimeException or Error

A Tale of Two Choices

- ▶ Dealing with checked exceptions
 - Can **propagate** the exception
 - Just declare that our method will pass any exceptions along
 - **public void loadGameState() throws IOException**
 - Used when our code isn't able to rectify the problem
 - Can **handle** the exception
 - Used when our code can rectify the problem

Handling Exceptions

- ▶ Use try-catch statement:

- `try {`
 - `// potentially “exceptional” code`
- `} catch (ExceptionType var) {`
 - `// handle exception`
- `}`



Can repeat this part for as many different exception types as you need.

- ▶ Related, try-finally for clean up:

- `try {`
 - `// code that requires “clean up”`
- `} finally {`
 - `// runs even if exception occurred`
- `}`

Vector Graphics Assignment

- »» A team project to create a scalable graphics program.

Vector Graphics Teams – Clifton

1 1 ,addantnb,filhobc,taos,wilsonam

1 2 ,caijy,चना1 ,lyonska,spurrme

1 3 ,hopwoocp,luok,nelsonca,shanx

1 4 ,cornetcl,eckertz,elswicwj,hirtjd

Team number used in repository name:

<http://svn.csse.rose-hulman.edu/repos/csse220-201130-vg-teamXX>

Check out *VectorGraphics* from SVN

Vector Graphics Teams – Defoe

21,finnelhn,haydr,lawrener

22,oliverr,senatwj,myersem,rybickcb

23,cooperdl,fengk,amesen

24,solorzaa,whitemrj,hazelrtj,zhenw

25,mehrinla,tilleraaj,vassardm

Team number used in repository name:

<http://svn.csse.rose-hulman.edu/repos/csse220-201130-vg-teamXX>

Check out *VectorGraphics* from SVN

Teaming

- ▶ A team assignment
 - So **some division of labor is appropriate** (indeed, necessary)
- ▶ A learning experience, so:
 - Rule 1: ***every* team member must participate in *every* major activity.**
 - Rule 2: **Everything that you submit for this project should be understood by *all* team members.**
 - Not necessarily all the details, but all the basic ideas

Work time now

- ▶ Read the specification
- ▶ Exchange contact info – you may want to add to your planning folder.
- ▶ Start working on your milestone 0 **due tomorrow**
 - Try to get it done in class today so you can:
 - Get some feedback in class before it's graded.
 - Focus on studying for the exam tonight.

Plan, then do

- ▶ If you complete these, show me:
 - CRC cards
 - UML – as complete as you can – will help coding later.
 - User stories for cycle 1
- ▶ Ask questions as needed!
- ▶ Work on the rest, cycle one due Thursday.
 - There's a quick partner eval
 - At end of project
 - If you are done, you have my blessing to start coding cycle 1
 - Use any reasonable combination of:
 - group meetings and/or
 - dividing up the work