Midterm style questions

- Multiple choice (choose one answer out of 4 answers)
- Fill in the blanks, such as:
  - What is the output of the following piece of code
  - Circle any syntax/compile/run time errors in the following piece of code and describe the problem
  - Fill in the names of terms (given the definition)
- True/False (justify your answer)
- Short answer:
  - Couple of sentence answer (example: explain how interfaces can be used to provide multiple inheritance in Java)
  - Provide a definition for the following term (e.g. encapsulation)
  - Write a short piece of code to do the following
- The midterm will be marked out of 100 marks (therefore 2 marks per minute – on average!)
- Closed books, no notes, no calculators.
- Arrive early, you will need to spread out in the room.

Midterm advice

- Review posted notes, and chapters 1 and 2
- Assignment 1 review (if you had problems, try to work through them this weekend)
- Write lots of small code examples testing the features we have discussed, draw pictures to describe “what is going on” in your programs
- Test yourself by asking questions about each of the slides, and in particular review the quizzes we did in class
- Find a study partner for the weekend! Test each other! Write snippet of code and try to trick each other with hard questions!
- Use the Web Forum to post “potential” midterm questions for others to consider! Then follow up with the answers!

Midterm hints

- Everything we have discussed so far (including today)
- Some example questions:
  - Why does OO design make it easier to reuse code?
  - Define the following terms (e.g. encapsulation, polymorphism, coupling, cohesion)
  - Order of constructors in constructor chaining, how does this occur?
  - What is the difference between composition and inheritance?

Key things to review

- Upcasting/downcasting of objects
- Casting between primitive types
- Static methods and fields
- Passing objects (and variables) by reference
- Inheritance
- Interfaces
- How to debug/comment programs
- Identifiers and reserved words, naming conventions
- Creating objects
- Arrays
- Control flow features
- Operators
- Strings
- Class modifiers
- Field and method modifiers, usage modifiers
- CRC cards
- Overriding, overloading, hiding, extending
- Exceptions
- Polymorphism