

COSC 121: Computer Programming II

Dr. Bowen Hui
University of British Columbia
Okanagan

A3 Review

- Common mistakes for Q1:
 - Not adding new tests as required
 - Not covering all cases in list methods
 - Not doing the test class with user input
- Common mistakes for Q2:
 - Not adding damage to score in attack()
 - Not checking if the animal has fainted before it has attacked, allowing attacks from the grave
 - Not sorting or searching properly
- There were some really cool and creative implementations of Q2! Kudos to those who went above and beyond to really customize the solution and make it your own!

Things to Bring

- Allowed:
 - Two pencils
 - An eraser
 - Two pens
 - One sheet of 8.5" x 11" paper
- Not allowed:
 - No calculators of any kind
 - No cell phone
- Required:
 - Student ID card

Format

- Note: This is a 2-hour exam
- Date and location on UBCO website

Instructions

- We will place the exam booklets face down on specific seats
- The exam starts only when we indicate so
 - If anyone starts earlier, the case will be treated as cheating
- The exam ends when we indicate “time’s up”
 - If anyone stops later, the case will be treated as cheating

Instructions (cont.)

- No looking around, whispering, using any notes/books aside from the cheatsheet, copying from others, listening to audio, using the cell phone for any purpose
 - If any such behaviour is found, the case will be treated as cheating
- If you have a question, raise your hand and ask one of us!

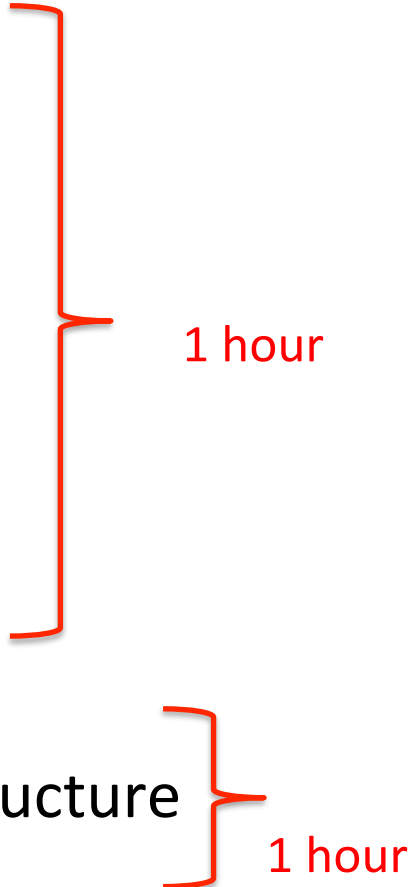
Instructions (cont.)

- When you are done:
 - Show your student ID + sign out
- Remember:
 - Final exam is worth 40% of grade
- Missed exam will receive 0 unless a medical reason is accepted by the Dean's Office

Format

- Multiple choice (10 points)
 - Mixed of topics
- Short Answers (25 points)
 - Working with abstract/concrete classes
 - Fixing polymorphism code
 - File and exception handling code
 - Sorting – show all intermediate steps
 - Tracing recursive method
- Writing Code (25 points)
 - Sorting or searching with linked list data structure

Format

- Multiple choice (10 points)
 - Mixed of topics
 - Short Answers (25 points)
 - Working with abstract/concrete classes
 - Fixing polymorphism code
 - File and exception handling code
 - Sorting – show all intermediate steps
 - Tracing recursive method
 - Writing Code (25 points)
 - Sorting or searching with linked list data structure
- 
- 1 hour
- 1 hour

Questions?