Name: _____

Student ID: _____

Lab Section: _____

– You have 60 minutes to write the 5 questions on this examination. A total of 50 marks are available.

– You are allowed **one** 8.5 X 11 two-sided sheet of handwritten notes. No other notes, books, calculators, computers, CD players, walkmans, boom boxes, robots, walkie-talkies, carrier pidgeons, nulcear weapons, or cellular phones are allowed.

– The number in square brackets to the left of the question number indicates the number of marks allocated for that question. Use these to help you determine how much time you should spend on each question.

– **Justify all your answers**

– Use the back of the pages for your rough work.

– **Good luck!**

Question	Marks
1	
2	
3	
4	
5	
Total	

UNIVERSITY REGULATIONS:

– No candidate shall be permitted to enter the examination room after the expiration of one half hour, or to leave during the first half hour of the examination.

– CAUTION: candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.

  1. Making use of any books, papers or memoranda, electronic equipment, or other memory aid devices, other than those authorised by the examiners.

  2. Speaking or communicating with other candidates.

  3. Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.

– Smoking is not permitted during examinations.

[10] 1. Short Answers

[2] a. *Briefly* describe the difference between a `class` and an `object`.

[2] b. The first step in the Object-Oriented design process is to *discover the objects*. Explain briefly what this means.

[6] c. For each of the following, describe a situation or give a reason why the first data structure mentioned would be preferred over the second. For example, *list > vector* means describe a situation when or give a reason why a list would be preferred over a vector.

    [2] i. *list > vector*

    [2] ii. *queue > list*

    [2] iii. *vector > list*

[10] 2. You have been asked to design a solution for the following domain:

A computer game involves a hero, some treasures and some monsters each of which is at a certain position. The hero moves around trying to find the treasures, and the monsters move around trying to eat the hero. There are three kinds of monsters: dragons, dopplegangers, and berserkers. The hero and the monsters all have a life-force. The hero is looking for gold, a particular kind of treasure Other treasures are weapons or ammunition that the hero can use to vaporize the monsters. There are three kinds of weapons: the ray blaster, the slingshot, and the magic cloud. Each weapon takes a different kind of ammunition.

Draw a picture showing the different classes (objects) in this domain, and how they relate to one another using has-a and is-a relationships.

[12] 3. A video game involves monsters, each of which has a life-force (whole number $>= 0$) which describes how healthy the monster is, and a strength (whole number $>= 1$). A monster is alive if its life-force is > 0, and it is dead otherwise. A monster can be queried as to its status (alive or dead), but keeps its life-force and strength values secret otherwise. A monster (the attacker) can attack another monster (the attacked). The winner of an attack is the monster with the greater strength. In cases of equal strength, the attacker wins (the attacked loses). The loser in an attack gets 1 life-force point deducted, while the winner's life-force stays the same.

[8] a. Design a class for monsters in this game according to the specification given, and write the class declaration. Your class should support constructing default monsters, and constructing monsters with a given initial life-force and strength. Don't worry about include statements.

[4] b. Write the implementation of all your methods declared in part (a).

[8] 4. The *length* of a vector is the square root of the sum of the squares of its elements. **For example**, if the vector is $\{1.1, 3.2, 2.7\}$, the length is $\sqrt{1.1^2 + 3.2^2 + 2.7^2}$. Write a procedure `length` which computes the length of a vector. The declaration is You can use the function `sqrt`, whose delcaration is `float sqrt(float);` which returns the square root of its argument.

[10] 5. A simple way to reverse a sequence is to push the elements of the sequence onto a stack, and then pop them off the stack, adding them to the new (reversed) sequence. Write a function `int reverse(int);` whose result is the input integer argument reversed. Your function **must** use a stack to do the reversal. That is, `y = reverse(493289);` will assign 982394 to `y`.