Family Name: ________________________ **(As you are sorted in the university registration system)**

Other Names: ____

Student Number: ___________

Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa

L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

SEG 2105
Introduction to Software Engineering

FINAL EXAMINATION –Sample

Time Allowed for Examination: Three hours

Professor: Timothy C. Lethbridge

Open book. (you may use books, personal notes etc., but no electronic devices)

Good luck!

For the multiple choice questions, **<u>Circle</u> the single best answer**. There is **only one best answer** for each multiple choice question. Part marks *may* be given for answers which are good but not the best.

Questions 1-5 are by default worth 2 marks each (total of 10 marks)

If you are unsure about an answer, mark it with '??'. **In question 6 you will be asked to indicate which of your answers you think you are most likely to have got wrong**.

1. Which of the following patterns describes a precise way to write code?
   a) Abstraction-occurrence
   b) Delegation
   c) Pipe-and-filter
   d) Singleton
   e) More than one of the above

2. In user interface design, the word 'affordance' means?
   a) The set of tasks the user knows how to do
   b) The set of features the customer has the money to pay for
   c) The set of capabilities that a particular screen or dialog allows the user to do
   d) The set of guidelines the user interface designer should follow in user-centered design
   e) None of the above

3. Which of the following is true?
   a) Most testing is done top-down
   b) Big-bang testing is best because it gives results faster.
   c) Most testing is done using a hybrid between top-down and bottom-up.
   d) It is not worth doing bottom-up testing because it requires writing a lot of drivers.
   e) Bottom-up testing means testing the user interface first.

4. Common coupling is named from:
   a) The term used in Cobol for calling a procedure
   b) Organizing all similar entities into one common location
   c) The fact that it is the most frequent type of coupling
   d) The term used in Fortran for global variables
   e) An approach where the code is owned by everybody (i.e. 'in common')

5. Which of the following is not part of the software engineering code of ethics?
   a) Participate in lifelong learning
   b) Act consistently with the public interest
   c) Maintain integrity and independence
   d) Do what your management tells you to do
   e) Be fair and supportive of colleagues

6. Indicate below the number of the single question from 1 to 5 that you think you are most likely to have got wrong. If you did indeed get it wrong, you will not lose marks for that question. If you indicate more than one number, then what you indicate will be ignored.

The single number of the question from 1-5 you think you are most likely got wrong is: _____

Short answer:

7. (5 marks) Describe the reasons why it would not be a good idea to modify the code in a framework like OCSF when you are using the framework in your application? Bug fixes and updates by others would not find their way to your code. Likely to introduce bugs; maker of framework can't help.

8. (5 marks) You are creating a method in Java that takes the string for a University of Ottawa course taught in English (such as "SEG2105") and returns the equivalent course taught in French ("SEG2505" in this case). (4 is added to the second digit). The method throws an exception if the input isn't a code for a course taught in English. Describe a set of equivalence classes to test this system.

- Non-English Course input
  - Not 3 letters
  - Not 4 numbers
  - Second digit 0
  - Second digit 5-9
- Second digit 1-4 -> Correct output

9. (5 marks) Draw a state machine that has the following States A, B, C and D. D is a superstate, where B and C are its substates. A and C are default start states. Event e1 causes a transition from any state to A. Event e2 causes a transition from C to D and vice-versa, and event e3 causes a transition from A to D.

10. (4 marks) In what kind of diagrams can a generalization triangle appear?
Use case, class – for full marks. Also component and others

11. (3 marks) What techniques that we discussed in this course involve a moderator?
Brainstorming, inspection

The following system description forms the basis for the remaining questions:

You are starting an outdoors services business. In the summer your employees will be mowing lawns and in the winter they will be clearing snow. All year round they will be trimming trees.

Some employees can do all three tasks; however since trimming trees requires specialized training, not all employees will be able to do it.

Customers will be signing contacts with your company for each of the tasks. For snow clearing you will be charging $1 per square meter of snow cleared per visit, and for lawn mowing you will charge 5 cents per square meter. You give a 50% discount for people willing to sign up for regular snow clearing or lawn mowing for an entire season.

Each day you dispatch crews to do the required work. Employees have to sign out snow plow trucks, snow blowers, shovels, lawn mowers and tree trimming kits. Tree trimming is always done in crews of two people; snow clearing is always done individually.

ANSWER ALL OF THE FOLLOWING IN YOUR EXAM BOOKLET.

12. (14 marks) Create a class diagram for the above system See end

13. (4 marks) Write in OCL a constraint to say that if the work is tree trimming, then the crew size must be at least 2.

Context SpecificTask inv:
  contract.type = “snow” implies dailyCrewWork.Employee -> size() > 1

Umple for this:

http://cruise.site.uottawa.ca/umpleonline/umple.php?model=122

```
class Employee {
  name;
  Boolean treeTrimCertified;
}

class Customer {
  name;
  address;
}

class Contract {
  Date startDate;
  Date endDate;
  1..* -- 1 Customer;
  contractType;  // {snow { } tree { } grass { } }
  Integer area;
}


class SpecificTask {
  Date taskDate;
  * -- 1 Contract;
  Float amountBilled;
  Time timeCompleted;
}

class DailyCrewWork {
   Date workDate;

   // an employee is can be on many crews, but one per day
   * -- 1..* Employee crew;
   1 -- 1..* SpecificTask work;
   * -- * PieceOfEquipment equipmentSignedOut;
}

class EquipmentType {
  description;
}

class PieceOfEquipment {
  Integer idNumber;
  * -- 1 EquipmentType;
}//$?[End_of_model]$?

class Employee
{
  position 50 30 109 45;
}

class Customer
{
  position 277 444 128 80;
}

class Contract
{
  position 58 435 129 97;
  position.association Contract__Customer 163,70 0,59;
}

class SpecificTask
{
  position 39 285 170 97;
  position.association Contract__SpecificTask 130,97 56,0;
}

class DailyCrewWork
{
  position 60 170 131 63;
  position.association
DailyCrewWork__PieceOfEquipment:equipmentSignedOut 132,28 0,10;
  position.association DailyCrewWork__SpecificTask:work 24,63 41,0;
  position.association DailyCrewWork__Employee:crew 30,0 34,80;
}

class EquipmentType
{
  position 273 287 149 63;
}

class PieceOfEquipment
{
  position 276 143 149 63;
  position.association EquipmentType__PieceOfEquipment 31,63 33,0;
}
```