

## George Mack, Computer Science – WID Teaching Portfolio

### F. Sample Assignments Produced and Carried Out Under the Writing Fellowship

Course: 420-315-DW, Programming IV – Advanced C# and .NET Programming

#### F-2. Sample assignment: In-Class Writing and Reflection

I carried out a number of assignments like this in two different courses. Here is a single example from course 420-315, given Monday February 7, 2011 (course week 3).

Objectives: (a) (As part of learning to work in a given programming language), Learn to use the available documentation resources to research programming methods and useful constructs, i.e. Carry out information research related to the assigned programming task; (b) Learn to distinguish between good and bad information sources; (c) Apply critical thinking; (d) Recognize ethical issues relevant to the field of study; (e) Analyze and produce various forms of discourse; (f) Communicate in the forms of discourse appropriate to (the) field of study, using all four modes of language i.e. reading, writing, listening and talking; (g) organize and use data (specifically, currency values).

Competencies developed: 0004, 000F, 000L, 000U, 016T, 016U, 016V, 0170, 017A, 017C, 017D.

Methodology:

Preparation:

Introduce the subject of data representation in the C# language. After covering basic scalar data types, discuss/demonstrate inherent floating point numerical errors using an interactive C# demo program.

Mention briefly the dangers of using floating point numeric types to represent currency values (loss of precision = inability to balance accounts). Mention other programming languages which did or did not solve this problem by providing specialized internal data types (COBOL, Visual Basic, Java...).

In-class Activity:

Have students download to their laptops a PDF file containing the Web article “Money Type for the CLR” from <http://moneytype.codeplex.com> (saved at

[http://dept.dawsoncollege.qc.ca/compsci/gmack/info/Article\\_moneytype.codeplex.pdf](http://dept.dawsoncollege.qc.ca/compsci/gmack/info/Article_moneytype.codeplex.pdf))

Have students read this article individually, and write (using Microsoft Word) a synopsis of what they learned from it (10 minutes, not to exceed 100 words). In particular, answer the question “What’s the right way to store money amounts in C#?”

Have students combine into small groups (of four or so), read their writing to their group, and discuss to understand, then each group to create a synthesized group article.

Have two or three groups read their article to the class, and provide instructor feedback.

(Homework activity) Have each student revise their own synopsis and email it to me (check-off evaluation).

Follow-On Activity (Thursday February 10, 2011):

Explain that the .NET CLS Decimal data type may be used to accurately represent a currency amount, but that the type of currency is not stored. Have students carry out an assignment to develop a Currency class in C# that can be used to represent currency amounts, including magnitude and type.

#### **F-4. Sample assignment: Peer Teaching Activity**

Objectives: Based on work reported in (Welch, Gradin, & Sandell, 2002) (a) teach the C# language-related particularities of defining class methods and behaviours, including scope rules and access modifiers; value, reference and output parameters; named, optional and default parameters; scalar and structured return types including arrays, objects, structs and tuples; (b) carry out a jigsaw-type cooperative learning exercise as a substitute for traditional lecturing; (c) give students practice with researching in the MSDN Help Library documentation for the C# programming language.

Competencies developed: 0004, 000F, 000L, 000U, 016N, 016T, 016U, 016V, 016Z, 0170, 0171, 0172, 0173, 017A, 017C, 017D.

Methodology:

This activity took place on March 21st and 24th 2011.

Preparation:

Instead of arriving at class with 40-50 PowerPoint slides and lecturing on the topic, arrive with only the table of contents slide for the lecture. (NB the instructor still needs to know the material to be covered!)

In-class Activity:

Review the table of contents for the lecture, then explain how the peer teaching activity works.

Provide the class with web hyperlinks to the required reference materials on the MSDN site.

Break students into groups of four or five and assign portions of the lecture materials to be developed by each group. Allow 30 minutes for students to learn the material, explain it to each other, and prepare PowerPoint slides containing outline points and sample C# code. While students are working in small groups, circulate around the room, listening to student interactions, answering questions and providing guidance to keep students on track, productive and within scope.

Have student groups present their material to each other (45 minutes allocated, this time was not enough. Some material spilled over into the following lecture period).

Follow-On Activity:

The subsequent lab assignment required students to use this material in programming work. Student PowerPoint presentations were combined by the teacher uploaded to the course server so that they would be accessible to all students.

#### **F-5. Sample assignment: Project-Based Cooperative Learning Through Jigsaw Assignments**

This activity took place over a ten week period, weeks 5 through 15 of the semester, from February 24 to May 12, 2011. In addition to classroom and lab activities, a field trip was organized on April 14 2011 to give students direct experience with project subject matter (Koi fish) and with examining the information systems of an external organization.

Objectives: (a) use an object-oriented development environment; (b) develop and design applications in a graphical environment; (c) organize data; (d) develop conceptual models using a structured approach; (e) produce and manage design documentation; (f) analyze, design, program and implement an application; (g) carry out all objectives in a collaborative, jigsaw-style project structure, which implies interacting and communicating in various work situations by verbal and written means, and learning various workplace functions of a programmer-analyst. References: (Microsoft Corporation, 2010), (Jacobson, Booch, & Rumbaugh, 1998)

Competencies developed: 0004, 000F, 000L, 016N, 016T, 016U, 016V, 016Z, 0170, 0171, 0172, 0173, 017A, 017C.

Methodology:

Preparation:

A suitable project to be carried out for an external organization was identified. This was the development of a DBMS-based information system to support the operations of Koi (fish) Competitions. Obtain the necessary information from actual users (organizers and judges).

In-class Activities:

All project activities were carried out in small groups in class and lab periods. The groups varied in size from pairs of two (for most programming and some design and writing activities), to groups of five or six members for certain design tasks. Larger groups were expected to manage work subdivision internally, and did so with varying degrees of success. Membership of the larger groups was shuffled twice during the semester to provide additional experience in working with different types of people. I, the teacher, acted as project manager and customer liaison.

Students were expected to produce deliverables each week, including design documents and source files (from Microsoft Visual Studio 2010 Ultimate Edition design tools), data files and C# programs (solutions) in source files.

Among the tasks, the students:

- Analyzed the features of the customer information system by reading customer produced documents and experimenting with their existing information system (to be replaced);
- Carried out a brainstorming activity and produced concept map diagrams;
- Produced use case diagrams in groups and compared them as a class to other groups' solutions;
- Designed and documented competing OO and relational data models, then worked in new groups to analyze and combine the various data models;
- Developed data dictionaries for two different database management systems (Oracle and Access);
- Developed test data for the two target data base systems;
- Created data bases, tables, relations and uploaded test data;
- Designed user interfaces (forms) and specified related data validation requirements;
- Coded and implemented user interfaces to input, store, update, delete and display data.

Follow-On Activity:

The course midterm and final exams assessed students' knowledge in these areas. The completed system components will become part of a finished system to be delivered to the customer.

### **F-6. Sample assignment: Exam Essay Question**

Objectives: (a) Give the students additional homework practice in writing about computer science; (b) provide learning activities about the workplace functions of a programmer/analyst, specifically the design functions described by competencies 016T, 0170, 0173, 017C and 017D (or A) and encountered this semester for the first time, as opposed to the programming and debugging functions described by those same objective but learned and perfect over semesters 1 through 4; (c) assess students' understanding of these concepts and their writing in the discipline by having them write a short essay on their final exam.

Competencies developed: 0004, 000L, 016N.

Methodology:

Preparation:

Before the last class, create an exam essay question (see below) that covers material that has already been learned about the design and development tasks of a programmer-analyst.

In-class and Final Exam Activity:

In class, one week before the final exam, provide the students with the text of an essay question that will be found on the final exam.

Have the students discuss possible answers to the question in small groups.

Ask a few groups to present a verbal synopsis of their group answer, and provide feedback and guidance.

Inform students that they will be expected to answer this question from memory during the final exam.

Administer the exam question and mark it. (NB: Rubric to be developed: this time the marking was just pass/fail; the object for the instructor was to evaluate what types of response can be expected from such an activity.) In addition, all student answers were digitized and saved to disk for further analysis at a later date.

Exam Question (worth 15 marks out of 100):

Write 150 to 200 words on the following topic.

Thesis: There is a significant difference between the tasks of (a) programming and (b) system analysis and design. Learning to program is not the same as learning to analyze and design. Different skills are required. In this course we have done both tasks...

Discuss what skills we learned and used for analysis and design (think about our class and lab activities). Discuss what skills a good programmer-analyst should have. Agree or disagree with the thesis, and give reasons.