Concordia University
Department of Computer Science
And Software Engineering

Home Examination

Course: COMP 6411
Instructor: Dr. Joey Paquet
Issued: Wednesday, February 27th, 2013
Due: Wednesday, April 3rd, 2013

General instructions and information

- The value of all questions is provided, as well as an approximation of the expected length of the answer to be provided.
- This is an open-book examination.
- For all answers, strong and numerous references must be provided.
- This is an ***individual*** examination, plagiarism will result in strong mark deductions and even failure.
- Answer ***question 1 and 2 and a total of 50% among questions 3-7***.
- A bibliography must be included separately with the answer for **each** answered question. References should be given within the text of your report, using a standard referencing style such as [Paquet 2010], [Paq10], or [1]. If you quote verbatim from a source, make it clear by using quote marks, italics font, or for longer quotes, putting the quote in an indented paragraph on its own. References that are not explicitly referred to in the text will not be counted.
- The report will be graded on factors including the quality of your answers, quality of language used, quality of presentation, and quality of references.
- Your report should be submitted electronically as a ***pdf*** file to the EAS as *theory_assignment-1* before midnight on Wednesday, April 3rd, 2013. Late submissions will not be accepted.

A. Mandatory questions (50%):

1. **[25%: 2-3 pages]** Define what is polymorphism and its different forms. Explain how it is related to the notion of "dynamic binding" in C++ implementation. Explain what is the "slicing problem" in C++ implementation, and explain how polymorphism is related to it. Explain why "overriding polymorphism" and "overloading polymorphism" are generally not considered true polymorphism.

2. **[25%: 2-3 pages]** In the history of programming languages, there has been languages that were "breakthrough" languages, i.e. languages that pioneered some or many features that later became part of many programming languages. Select one such breakthrough language, describe the history of its creation, and explain what breakthrough features it introduced, why these features were important, and what subsequent important languages used or improved on these features.

B. Choose for a total of 50% among the following questions:

3. **[20%: 2-3 pages]** Describe what are *static* (i.e. compile-time) and *dynamic* (i.e. run-time) type checking. Compare their advantages and disadvantages from the point of view of both the language designer and the programmer. Discuss reasons why using dynamic typing in a language's implementation may limit the language's use in some specific application areas.

4. **[20%: 2-3 pages]** In the history of programming languages, there has been a variety of procedure call parameter passing mechanisms. ALGOL 60 provided call by value and call by name. Pascal provided call by value and call by reference. ALGOL-W used, among others, call by result and call by value-result. Briefly describe the calling mechanisms just mentioned and discuss why most modern programming languages provide only call by value.

5. **[20%: 2-3 pages]** Outline the key features that a language must have to be called object-oriented and briefly discuss to what extent C++ and Java have them. Briefly discuss some of the reasons why C++ programs would typically run faster than equivalent programs written in Java. Explain what improvements were made to Java that explain why Java programs are now, in many cases, executing as fast as C++ programs.

6. **[10%: 1-2 pages]** Identify non-von Neumann computing models and explain their mode of functioning. Give examples of programming languages adapted to these alternate computing models.

7. **[10%: 1-2 pages]** Discuss and evaluate the differences in the typing disciplines used by Python and C. Compare the advantages and disadvantages that their designs impose on the programmer.