Midterm Preparation
CS 2ME3/SE 2AA4

Steven Palmer

Department of Computing and Software
McMaster University

February 13 - 17

Outline

Midterm Format

- 90 minutes
- Multiple choice questions
- Written notes, printed notes and textbook are permitted.
- All electronic devices are **not** permitted.

What Material Should I Review?

Everything covered during class and tutorials is fair game. You should review:

- All lecture slides
- Ghezzi: Ch. 1 to 4
- Hoffman & Strooper:
    - Explicitly covered: Ch. 3, 6 and 7
    - Implicitly covered: Ch. 1 and 2 (overlaps with Ghezzi)
- Parnas papers in repository
- Tutorial slides
- Assignments and solutions

Example Q1

External software qualities are the qualities that result from the use
of external libraries in software code.

1 True
2 False

Example Q1 Answer

External software qualities are qualities that result from the use of external libraries in software code.

1 True
2 **False**

Example Q2

A _____ is a general concept that is widely applicable in software engineering.

1. tool
2. technique
3. principle
4. method

Example Q2 Answer

A _____ is a general concept that is widely applicable in software engineering.

1. tool
2. technique
3. **principle**
4. method

Example Q3

A _____ system enables reasoning to be mechanized.

1. rigourous
2. modular
3. functional
4. formal

Example Q3 Answer

A _____ system enables reasoning to be mechanized.

1. rigourous
2. modular
3. functional
4. **<u>formal</u>**

Example Q4

In the Module Guide (MG), which of the following does not fit the criteria for a good module secret?

1. one module should have one or more secrets
2. secrets should often be nouns
3. secrets are often phrased as "How to..."

Example Q4 Answer

In the Module Guide (MG), which of the following does not fit the criteria for a good module secret?

1. **one module should have one or more secrets**
2. secrets should often be nouns
3. secrets are often phrased as "How to..."

Example Q5

Given the following MIS access routine semantics specification for a function f,

$f(x, y)$:

- output: $out := \sqrt{x^2 + y^2}$
- exception: none

What do we know about how f(x,y) is implemented?

1. a math library is required to perform the square root operation
2. the function consists of a single line that calculates and returns the output
3. both 1 and 2
4. nothing

Example Q5 Answer

Given the following MIS access routine semantics specification for a function f,

f(x, y):

- output: $out := \sqrt{x^2 + y^2}$
- exception: none

What do we know about how f(x,y) is implemented?

1. a math library is required to perform the square root operation
2. the function consists of a single line that calculates and returns the output
3. both 1 and 2
4. <u>nothing</u>

Example Q6

Which of the following is not true of pure functional programming
languages?

1. computation is treated as the evaluation of mathematical
   functions

2. functions have side effects

3. functions are a first order data type

Example Q6 Answer

Which of the following is not true of pure functional programming languages?

1. computation is treated as the evaluation of mathematical functions
2. **functions have side effects**
3. functions are a first order data type

Example Q7

```
def f(x,y):
  return x * y

numList = [2,4,6,8,10]

#BEGIN
product = 1
for e in numList:
  product = f(product, e)
#END
```

The code segment between the BEGIN and END comments is equivalent to which of the following?

1. product = map(f, numList)
2. product = filter(f, numList)
3. product = reduce(f, numList)

Example Q7 Answer

```
def f(x,y):
    return x * y

numList = [2,4,6,8,10]

#BEGIN
product = 1
for e in numList:
    product = f(product, e)
#END
```

The code segment between the BEGIN and END comments is
equivalent to which of the following?

1. product = map(f, numList)

2. product = filter(f, numList)

3. **product = reduce(f, numList)**

Example Q8

Git is an example of a _____ version control system.

1. distributed
2. centralized
3. local-only

Example Q8 Answer

Git is an example of a _____ version control system.

1. **distributed**
2. centralized
3. local-only