COP5025 Spring 2003 – Final Exam (Chs. 8+9+Prolog)

Name: ______________________________ (Please print)

Put the answers on these sheets. Use additional sheets when necessary. You can collect 100 points in total for this exam. A bonus question is included for an additional 15 points. **This exam is open book and open notes.**

1. **Prolog** (15 points) Consider the following assumptions made by Sir Bedever in *Monty Python and the Holy Grail* to argue that the girl is a witch:

```
witch(X)        :- burns(X), woman(X).
burns(X)        :- isMadeOfWood(X).
isMadeOfWood(X) :- floats(X).
floats(X)       :- sameWeight(X, Y), floats(Y).
floats(duck).
sameWeight(girl, duck).
woman(girl).
```

Mark the queries that return a successful answer:

	`floats(wood)`
	`floats(duck)`
	`floats(girl)`
	`burns(duck)`
	`burns(girl)`
	`witch(duck)`
	`witch(girl)`

2. **Propositions** (10 points) Consider the language of propositional formulas (lecture note 94 and Section 8.5). Given $\rho = \{\langle x, \mathrm{TRUE}\rangle, \langle y, \mathrm{FALSE}\rangle, \langle z, \mathrm{TRUE}\rangle\}$, which of the following formulas is true? Show how you derived your answers.

   (a) $\mathcal{M}[\![x \Rightarrow y]\!]\ \rho$

   (b) $\mathcal{M}[\![\neg z \Rightarrow (y \Rightarrow x)]\!]\ \rho$

3. **Expressions** (10 points) Consider the language of *simple expressions* (lecture note 95 and Section 8.6) What functions do the following expressions denote?

   (a) $1 - x$

   (b) **let** $y = x + x$ **in** $y - y$ **end**

4. **Ruby's unless** Ruby is a scripting language that features an "`unless`" construct for conditional execution:

```
x := 1 unless a
```

assigns 1 to `x` only when `a` is **NOT** true, i.e. unless `a` is true, `x` will be assigned 1.

   (a) (15 points) Give a denotational description of `unless` in the language of *combining expressions and commands* (notes 103–105 and Section 8.10). The condition is evaluated first after which the statement is conditionally executed (note that zero values are used to denote FALSE). Assume that `unless` returns 0 when the condition is true. You should properly deal with the side-effects (state changes) in your definition.

   $\mathcal{M}[\![E_1 \texttt{ unless } E_2]\!]\ \rho\ \sigma =$

   (b) (15 points) The proof rule for `unless` is:

   $$\frac{\{\neg B \,\&\, P\}\ S\ \{Q\} \qquad B \,\&\, P \Rightarrow Q}{\{P\}\ S \texttt{ unless } B\ \{Q\}}$$

   Prove that the following Hoare triple is valid:

   $\{y \leq 0\} \quad x := 1 \texttt{ unless } y > 0 \quad \{x = 1\}$

5. **Hoare triples** (15 points) Which of the following Hoare triples are valid?

   (a) $\{x = 2 \,\&\, y = 2\} \quad y := x \quad \{y = 2\}$

   (b) $\{P \,\&\, Q\} \quad \texttt{skip} \quad \{Q\}$ (for any P and Q)

   (c) $\{\text{TRUE}\} \quad \texttt{if } x \geq y \texttt{ then } x := x - y \texttt{ else } y := y - x \quad \{x \geq 0 \lor y \geq 0\}$

6. **Hoare triples with strings** (10 points) The concatenation ofa two strings s and t is denoted by $s//t$. Which of the following Hoare triples are valid? **Show how you derived your answer.**

   (a) $\{s = \text{“}\texttt{hello}\text{”} \,\&\, t = \text{“}\texttt{ world}\text{”}\} \quad s := s//t \quad \{s = \text{“}\texttt{hello world}\text{”}\}$

   (b) $\{t = \text{“}\texttt{abc}\text{”}\} \quad s := \text{“}\texttt{a}\text{”}//\text{“}\texttt{b}\text{”}//\text{“}\texttt{c}\text{”} \quad \{s = t\}$

7. **Procedure proof** (10 points) Consider the procedure

> **proc** `shift`**(in** x**, inout** y**, out** z**)**
> z **:=** y**;**
> y **:=** x**;**

and the procedure call:

> **call** `shift`**(0,** a**,** b**)**

Derive the weakest precondition of this call using the procedure proof rule:

$$\frac{\{P\}\ (z := y; y := x)\ \{R[(a,b) := (y,z)]\}}{\{P[(x,y) := (0,a)]\}\ \textbf{call}\ \texttt{shift}(0,a,b)\ \{R\}}$$

where R is the postcondition $a = 0 \,\&\, b = 1$.

Hint: first derive the weakest precondition P from the assignment statements and postcondition $R[(a,b) := (y,z)]$. Then apply the substitution $P[(x,y) := (0,a)]$ to find the weakest precondition of the procedure call.

8. **Bonus** (15 points)
Prove the correctness of Euclid's algorithm for the greatest common divisor:

z **:=** m**;**
r **:=** n**;**
while $r \neq 0$ **do**
    h **:=** $z\%r$**;**
    z **:=** r**;**
    r **:=** h**;**
end
$\{z = \mathrm{gcd}(m, n)\}$

The loop invariant I is:

$\mathrm{gcd}(z, r) = \mathrm{gcd}(m, n)\,\&\,(z \neq 0 \vee r \neq 0)$

Show how you derived the weakest precondition of the algorithm by annotating the program with pre- and postconditions for all statements. Use the following properties of *gcd*:

$$\begin{aligned} \mathrm{gcd}(z, 0) &= z \\ \mathrm{gcd}(z, r) &= \mathrm{gcd}(r, z\%r) \qquad \text{if } r \neq 0 \end{aligned}$$

where % denotes remainder.