



University of Arizona, Department of Computer Science

CSc 372 — Assignment 1 — Due noon, Wed Sep 8 — 3%

Christian Collberg  
September 1, 2004

## 1 Introduction

The purpose of this assignment is for you to become familiar with Scheme, DrScheme, and writing recursive functions.

Before starting this assignment, set your DrScheme language level to **Standard (R5RS)**.

All your function definitions should be *pure*, i.e. they should not use any of Scheme's imperative features such as `set!`. Also, never use iteration, always recursion.

Every function should be commented. At the very least, the comments should state what the function does, which arguments it takes, and what result it produces.

This assignment is graded out of 100. The Scheme assignments are worth a total of 10% of your final grade. This assignment is worth 3% of your final grade.

## 2 Simple functions

1. Using the formula

$$V = 2\pi rh + 2\pi r^2$$

define a function (`cylinder-surface-area r h`) which computes the surface area of a cylinder of height  $r$  and radius  $h$ : [10 points]

```
(define pi 3.14159265)

;; This function...
(define (cylinder-surface-area r h)
  ...
)
```

Your function should have the following behavior:

```
> (cylinder-surface-area 2 5)
87.96459420000001
```

2. Define a function (`double-string s`) which takes a string argument `s` and returns a new string consisting of two copies of `s`: [10 points]

```
;; This function...
(define (double-string s)
  ...
)
```

Your function should have the following behavior:

```
> (double-string "")
""
> (double-string "hello")
"hellohello"
```

3. Define a recursive function (`copy-string s n`) which returns a string consisting of `n` copies of the string `s`: [10 points]

```
(define (copy-string s n)
  (cond
    [(<= n 0) ...]
    [(= n 1) ...]
    [else ...]
  )
)
```

Your function should have the following behavior:

```
> (copy-string "hello" -1)
""
> (copy-string "hello" 0)
""
> (copy-string "hello" 1)
"hello"
> (copy-string "hello" 2)
"hellohello"
> (copy-string "hello" 10)
"hellohellohellohellohellohellohellohellohello"
```

4. Define a recursive function (`power-of-two? n`) which returns `#t` if `n` is a power of two (i.e.  $n = 2^m$ ), and `#f` otherwise: [10 points]

```
(define (power-of-two? n)
  (cond
    ....
  )
)
```

Your function should have the following behavior:

```

> (power-of-two? 0)
#f
> (power-of-two? -4)
#f
> (power-of-two? 1)
#f
> (power-of-two? 2)
#t
> (power-of-two? 3)
#f
> (power-of-two? 4)
#t
> (power-of-two? 6)
#f
> (power-of-two? 8)
#t

```

### 3 Computing $\pi$

1. Define the factorial function (`! n`) which computes  $n! = 1 \cdot 2 \cdot \dots \cdot n$ : [10 points]

```

(define (! n)
  ...
)

```

2. Define the exponentiation function (`exp m n`) which computes  $m^n = \overbrace{m \cdot m \cdot \dots \cdot m}^n$ : [10 points]

```

(define (exp m n)
  ...
)

```

Compare the result against the built-in Scheme function (`expt m n`).

3. Use the functions you've defined above to compute  $\arctan(x)$  using the formula: [30 points]

$$\arctan(x) \approx \sum_{i=0}^n (-1)^i \frac{x^{2i+1}}{(2i+1)!}$$

Your function should take an extra argument `n`, the number of iterations to be used in the approximation:

```

(define (arctan x n)
  ...
)

```

Compare your result against the built-in function (`atan x`).

4. Using the formula

$$\frac{\pi}{4} = 4 \arctan\left(\frac{1}{5}\right) - \arctan\left(\frac{1}{239}\right)$$

define a function (`myPI`) which computes an approximation to  $\pi$ : [10 points]

```
(define (myPI)
  (exact->inexact ... (arctan 1/5 10) ... (arctan 1/239 10) ...))
)
```

The built-in function `(exact->inexact x)` converts an exact number to inexact (decimal) form. The output of your function should be similar to this:

```
> (myPI)
3.1619729398804797
```

## 4 Submission and Assessment

The deadline for this assignment is noon, Wed Sep 8. You should submit the assignment (a text-file containing the function definitions) electronically using the Unix command `turnin cs372.1 <files>`. This assignment is worth 3% of your final grade.

<b>Don't show your code to anyone, don't read anyone else's code, don't discuss the details of your code with anyone. If you need help with the assignment see the instructor or TA.</b>
--