COMP302: Programming Languages and Paradigms

Assignment Project Exam Help

Prof. Brigitte Pientka (Sec 01)
bpientka@cs.mcgill.ca

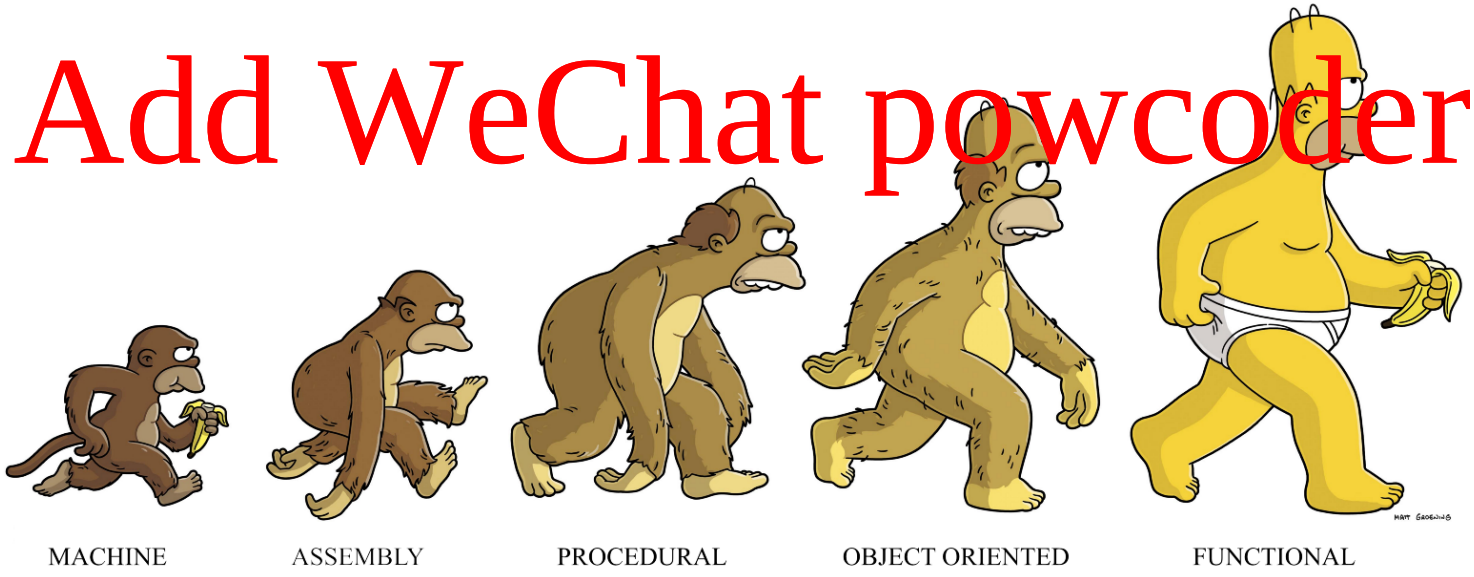
Francisco Ferreira (Sec 02)
fferre8@cs.mcgill.ca

https://powcoder.com

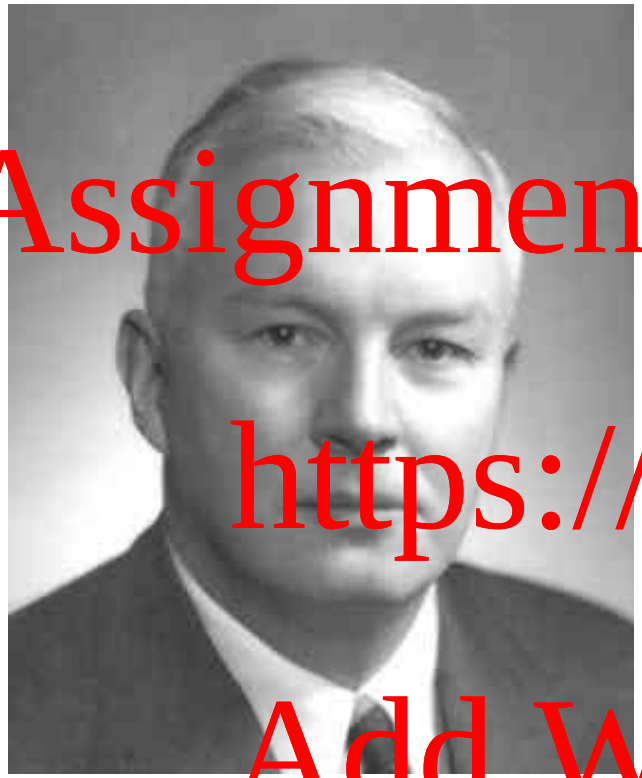
School of Computer Science
McGill University

Week 1-2, Fall 2017

Add WeChat powcoder



Functional Tidbit: Haskell B. Curry



- Logician and Mathematician
- 12 Sept 1900 – 1 Sept 1982
- Most known for the Curry-Howard-Isomorphism
  - Indirect relationship between programs and proofs
- Prog. language Haskell is named after him.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

What is OCaml?

Statically Typed Functional Programming Language

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

What is OCaml?

Statically Typed Functional Programming Language

- Types approximate runtime behaviour
- Analyze programs **before** executing them
- Find and fix bugs before testing

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

What is OCaml?

Statically Typed Functional Programming Language

- Types approximate runtime behaviour
- Analyze programs **before** executing them
- Find and fix bugs before testing

- Primary expressions are functions!
- Functions are first-class!
- Pure vs Not Pure
- Call-By-Value vs Lazy

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Concepts for Today

- Writing and executing basic expressions
- Learn how to read error message
- Names, Values, Basic Types
- Variable, Bindings, Scope of Variables
- Simple Functions

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

DEMO

Assignment Project Exam Help

https://powcoder.com

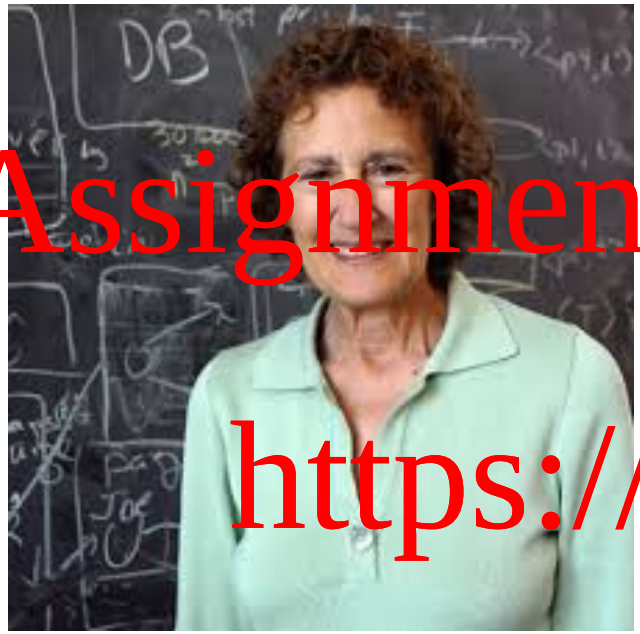
Add WeChat powcoder

Step 2:Variables and Bindings

- Variable binding is not an assignment
- Variables cannot be updated – we can only overshadow a previous binding
- Variable bindings persist
- Garbage collection disposes off variable bindings that are not needed anymore
- Variable bindings are local – they exist within a scope
- Variables are bound to a value – not an expression

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Function Tidbit: Barbara Liskov



- Professor at MIT
- John von Neumann Medal [2014]
- Turing Award for her work in the design of programming languages and software methodology [2008]

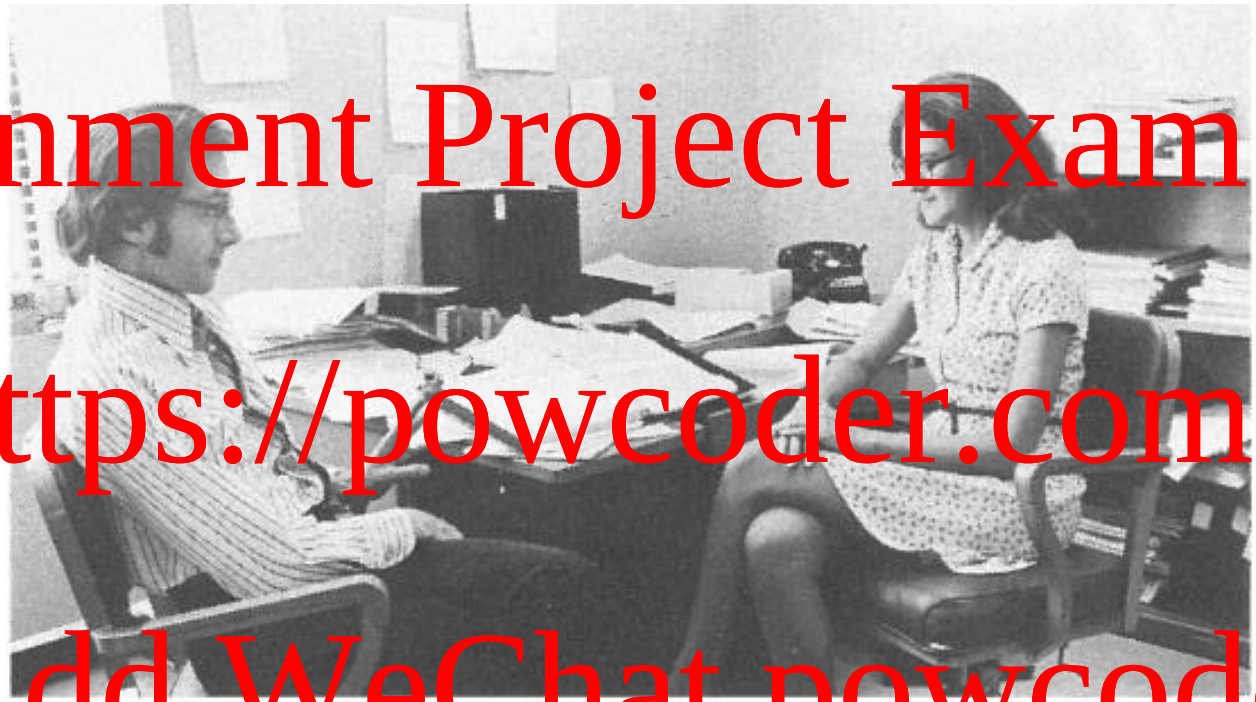
Assignment Project Exam Help
https://powcoder.com

"The motivation behind the work in very-high-level languages is to ease the programming task by providing the programmer with a language containing primitives or abstractions suitable to his problem area. The programmer is then able to spend his effort in the right place; he concentrates on solving his problem, and the resulting program will be more reliable as a result. Clearly, this is a worthwhile goal."

B. Liskov [1974]

Add WeChat powcoder

Function Tidbit: Barbara Liskov (in the 70's)



Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Functions

- Functions are values
- Function names establish a binding of the function name to its body

```
let area (r: float) = pi *. r *. r;;
```

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Functions

- Functions are values
- Function names establish a binding of the function name to its body

```
let area (r: float) = pi *. r *. r;;
```

- Recursive functions are declared using the keyword `let rec`

```
let rec fact n =
  if n = 0 then 1
    else n * fact (n-1)
```

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

DEMO

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Tail-recursive Functions

A function is said to be "tail-recursive", if there is nothing to do except return the final value. Since the execution of the function is done, saving its stack frame (i.e. where we remember the work we still in general need to do), is redundant.

Assignment Project Exam Help

https://powcoder.com

- Write efficient code
- All recursive functions can be translated into tail-recursive form!

Add WeChat powcoder

Tail-recursive Functions

A function is said to be "tail-recursive", if there is nothing to do except return the final value. Since the execution of the function is done, saving its stack frame (i.e. where we remember the work we still in general need to do), is redundant.

Assignment Project Exam Help

https://powcoder.com

- Write efficient code
- All recursive functions can be translated into tail-recursive form!

Add WeChat powcoder

Example: Rewriting Factorial

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Example: Rewriting Factorial

```
let rec fact_tr1 n =
  let rec f (n, m) =
    if n=0 then
      m
    else f(n-1, n*m)
  in
   f(n,1)
```

Assignment Project Exam Help
https://powcoder.com

- Second parameter to accumulate the result; in the base case we simply return its result
- Avoids having to return a value from the recursive call and subsequently doing further computation.
- Avoids building up a runtime stack to memoize what needs to be done once the recursive call returns a value

Add WeChat powcoder

Example: Rewriting Factorial

```
1 let rec fact_tr1 n =
2   let rec f (n, m) =
3     if n=0 then
4       m
5     else f(n-1, n*m)
6   in
7    f(n,1)
```

Assignment Project Exam Help
https://powcoder.com

- Second parameter to accumulate the result; in the base case we simply return its result
- Avoids having to return a value from the recursive call and subsequently doing further computation.
- Avoids building up a runtime stack to memoize what needs to be done once the recursive call returns a value

Add WeChat powcoder

What is the type of f?

Passing Arguments

- Passing all arguments at the same time

```
'a * 'b -> 'c
```

- Passing one argument at a time

```
'a -> 'b -> 'c
```

- **Remark:** We can translate any function of type `'a * 'b -> 'c` to a function of type `'a -> 'b -> 'c` and vice versa. This is called *currying* (*uncurrying* resp.)

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Assignment Project Exam Help

https://powcoder.com

Data Types and Pattern Matching

Add WeChat powcoder

Playing Cards

How can we model a collection of cards?

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Playing Cards

How can we model a collection of cards?

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Declare a new type together with its elements

```
1 type suit = Clubs | Spades | Hearts | Diamonds
```

User-Defined (Non-Recursive) Data Type

```
1 type suit = Clubs | Spades | Hearts | Diamonds
```

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

User-Defined (Non-Recursive) Data Type

```
1 type suit = Clubs | Spades | Hearts | Diamonds
```

- The order in which we declare these elements does not matter

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

User-Defined (Non-Recursive) Data Type

```
1 type suit = Clubs | Spades | Hearts | Diamonds
```

Assignment Project Exam Help

- The order in which we declare these elements does not matter.
- We call Clubs, Spades, Hearts, Diamonds constructors.

https://powcoder.com

Add WeChat powcoder

User-Defined (Non-Recursive) Data Type

```
1 type suit = Clubs | Spades | Hearts | Diamonds
```

Assignment Project Exam Help

- The order in which we declare these elements does not matter.
- We call Clubs, Spades, Hearts, Diamonds constructors.
- Constructors must begin with a capital letter in OCaml.

https://powcoder.com

Add WeChat powcoder

User-Defined (Non-Recursive) Data Type

```
type suit = Clubs | Spades | Hearts | Diamonds
```

Assignment Project Exam Help

- The order in which we declare these elements does not matter.
- We call Clubs, Spades, Hearts, Diamonds constructors.
- Constructors must begin with a capital letter in OCaml.
- Use pattern matching to analyze elements of a given type.

https://powcoder.com

```
match <expression> with
      | <pattern> -> <expression>
      | <pattern> -> <expression>
      ...
      | <pattern> -> <expression>
```

Add WeChat powcoder

A pattern is either a variable, underscore (wild card), or a constructor.

Comparing Suits

Write a function dom of type suit * suit -> bool

dom(s1,s2) = true iff suit s1 beats or is equal to suit s2 relative to the ordering

Spades > Hearts > Diamods > Clubs

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Comparing Suits

Write a function dom of type suit * suit -> bool

dom(s1,s2) = true iff suit s1 beats or is equal to suit s2 relative to the ordering

Spades > Hearts > Diamods > Clubs

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Demo