CPSC 418: Parallel Computing
In Class Worksheet

Winter 2016–2017 Term 2

Introduction to Erlang

Name: _____ Student Number: _____

Name: _____ Student Number: _____

Name: _____ Student Number: _____

1. **Read You Some Erlang.** What is the value of each Erlang expression? (Don't just type it in—try to reason out the answer.)

   (a) `42 =:= 6 * 7.0 or 1 <= 2.`

   (b) `tl([1|2]).`

   (c) `tl([1,[ 2, 3, 4 ]]).`

   (d) `length([ 1 | [ 2 | [ 3, 4, 5 ]]]).`

   (e) `x = [ 1 | [ [ 2, 3, 4 ], 5 ]].`

   (f) `[ { value, Purple } || Purple <- [ 5, george, "22" ] ].`

2. **Erlang Types.** What is the type of each of the following Erlang expressions (variable, atom, boolean, integer, float, list or tuple)?

   (a) `jeopardy`

   (b) `True`

   (c) `"3.14159"`

   (d) `6.02214e23`

   (e) `[ { ok, 42 } ]`

   (f) `false`

   (g) `_haberdashery`

   (h) `{ [ 1 | [ 2 | [ 3 ]]] }`

3. **Write You Some Erlang.** Write a function `sublist(List, Start, End)` where `List` is a list, `Start` is a positive integer and `End` is an integer greater than or equal to start. The function returns the sublist of `List` containing elements `Start` (inclusive) through `End` (exclusive). Remember that Erlang likes to start numbering elements from 1.

4. **Referential Transparency: Not Just for Functional Languages!** Write a function `dotProd(A, B)` in your favorite *imperative* language (eg: Java, Python, C, C++, etc.) which returns the dot product of two one-dimensional vectors of numbers. You may assume that `A` and `B` are stored in arrays, lists, or whatever data type is convenient for your language choice. However, *your code must display referential transparency*; in other words, you may not change the value of a variable once it is set.