

CSE 305 Programming Languages Spring, 2010

Homework 7

Maximum Points: 23

Due 10:30 AM, Friday, March 19, 2010

Professor Shapiro

March 5, 2010

Write the answers in a file named `hw7.txt`. Put your name and user name at the top of the file, and submit the file, using the UNIX command, `submit_cse305 hw7.txt`. You will be directed to submit 3 more files below.

1. (9) Do the following parts of exercise 7.9 (page 345–346) of the text. Instead of superscripts, write the integer to indicate order immediately after the right parenthesis. For example, the answer to

`a + b * c + d`

would be written as

`((a + (b * c)1)2 + d)3`

- (a) (3) `a * (b - 1) / c mod d`
- (b) (3) `(a - b) / c & (d * e / a - 3)`
- (c) (3) `-a + b`

2. (5) Fortran, Haskell, Python, and Ruby use `**` as an exponentiation operator.

- (a) (2) What is printed by the following Fortran program, which you may copy from `/projects/shapiro/CSE305/ProgramsForHomeworks/expTest.f`? (Programs compiled by `f77`, `f90`, and `gfortran` print the same number.)

```
Program exptest

Integer i, j, k

j = 5
k = 3

i = j ** 2 ** k - 1

Print *, i

End
```

- (b) (3) Show the order in which Fortran evaluates the expression, `i = j ** 2 ** k - 1`, using the same technique you did for Exercise 1, above.

3. (3) Fill in the blank in the Java program below, which may be copied from

```
/projects/shapiro/CSE305/ProgramsForHomeworks/Max.java
```

with an expression so that when you run the program you get the behavior shown below it. Don't add anything else to the program but the one expression. Submit your modified program as Max.java.

The program:

```
public class Max {  
  
    public static int max(int x, int y) {  
        return _____;  
    } // max  
  
    public static void main(String[] args) {  
        int x, y;  
        x = 3;  
        y = 5;  
        System.out.println("The maximum of " + x + " and " + y + " is " + max(x,y));  
        x = 5;  
        y = 3;  
        System.out.println("The maximum of " + x + " and " + y + " is " + max(x,y));  
        x = 5;  
        y = 5;  
        System.out.println("The maximum of " + x + " and " + y + " is " + max(x,y));  
    } // main  
} // Max
```

The run:

```
<timberlake:Test:1:152> java Max  
The maximum of 3 and 5 is 5  
The maximum of 5 and 3 is 5  
The maximum of 5 and 5 is 5
```

[Continued on next page.]

4. (3) Fill in the blank with a pattern to produce the behavior indicated in the rest of this example run of Erlang. Submit a one-line file containing your line “10>” assignment expression as hw7pattern.erl.

```
<timberlake:~:1:29> erl
Erlang R13B03 (erts-5.7.4) [source] [64-bit] [smp:2:2] [rq:2] [async-threads:0] [hi

Eshell V5.7.4 (abort with ^G)
1> [X1,X2] = [2,3].
[2,3]
2> X1.
2
3> X2.
3
4> [X3|X4] = [2,3].
[2,3]
5> X3.
2
6> X4.
[3]
7> [X5|X6] = [4,5,6,7].
[4,5,6,7]
8> X5.
4
9> X6.
[5,6,7]
10> _____ = [4,5,6,7].
[4,5,6,7]
11> X7.
4
12> X8.
5
13> X9.
[6,7]
14> q().
ok
15> <timberlake:~:1:30>
```

[Continued on next page.]

5. (3) Erlang is a functional programming language that uses strict evaluation. As discussed in lecture, one can get sequential behavior from a functional programming language via nested function calls. Consider the following Erlang program, which you may copy from

```
/projects/shapiro/CSE305/ProgramsForHomeworks/hw7a.erl

-module(hw7a).
-export([putFirst/2]).

putFirst(X,_) ->
    io:format("~w~n", [X]).
```

If you copy this program, and run the Erlang shell in the same directory as the program, you can do the following:

```
<timberlake:ProgramsForHomeworks:1:46> erl
Erlang R13B03 (erts-5.7.4) [source] [64-bit] [smp:2:2] [rq:2] [async-threads:0] [h

Eshell V5.7.4 (abort with ^G)
1> c(hw7a).
{ok,hw7a}
2> hw7a:putFirst(a,nil).
a
ok
3> _____
a
b
c
d
ok
4> q().
ok
5>
5> <timberlake:ProgramsForHomeworks:1:47>
```

What is the expression you can enter in place of the blank that uses `putFirst` and nested function calls to get the indicated behavior? Submit a one-line file containing that expression as `hw7call.erl`.