Wentworth Institute of Technology

COMP3350 Programming Languages

Homework 2

1. Modify the following F# function to use pattern matching instead of if-then-else. [2]

```
let result =
    if System.DateTime.Now.Second % 2 = 0 then
        "heads"
    else
        "tails"

printfn "%A" result
```

```
let result =
    match System.DateTime.Now.Second with
    | x when x % 2 = 0 -> "heads"
    | _ -> "tails"

printfn "%A" result
```

2. Given the function [4]

```
let add x y = x + y
```

Compute (add 4 6 8 10) using

i. Composition

```
let result = add (add 4 6) (add 8 10)
```

ii. Function chaining

```
let result = add 4 6 |> add 8 |> add 10
```

3. Do the following 2 functions produce the same result? Briefly describe what each function does? [4]

i.

```
let rec incrElements list =
    match list with
    | head :: tail -> head + 1 :: incrElements tail
    | [] -> []
printfn "Result = %A" (incrElements [4; 5; 6])
```

ii.

```
let result = List.map ((+) 1) [4; 5; 6]
    printfn "Result = %A" result
```

These two function in fact do the same thing, the first traverses the list recursively and adds 1 to each element as it comes across it. The second function applies a function to the entire list at once that adds 1 to every element in the list.