Principi di Linguaggi di Programmazione
Programming Paradigms

prof. M. Bellia
Mid Term Exam - March 4 2014

(Available Time: 2 hours)

Exercise 1 (pts 3) By using one of the following languages C, Java, Ocaml, write a program fragment that contains an example of aliasing. Then, uderline and comment the code that uses the aliasing.

Exercise 2 (pts 2) List the main features of inline blocks and of procedural blocks.

Exercise 3 (pts 2) Apply the Le Blank-Cook algorithm to the program below and replace each used identifier with the right pair.

```
{int x = 5;
 int z = 3;
 int p(int n){
        int z = n;
        x = x-3;
        if n>0 then z = n+p(x);
        return z;
        }
   {int x = 7;
    int y = 10;
    x = p(z);
    print(x+y);
   }
print(x);
}
```

Exercise 4 (pts 3) What is printed when dynamic scope is assumed for the program of the previous exercise?

Exercise 5 (pts 4)
(a) Complete the following definition for the sum of the first n natural numbers:

g ≡ Y G

G ≡ ... if(n=0) then 0 else n+f(n-1)

(b) Complete the extensional computation of YG(2) and show how the result has been obtained.

Exercise 6 (pts 5) Apply the formulas, given in the course, to the declaration below:

Mut final int x = 3 + y; final int y = 5 Ally

To do it:

(a) Correct: (1) the formula for g and (2) the formula for YH^0;

(b) Complete the text in order to compute the environment defined by the declaration.

$g \equiv Y...(...bind(y,...,\sigma)(\lambda\sigma.\lambda\mu.bind(x,..., \mu)(\mu)(\mu))(\mu))$

$H \equiv \lambda\mu.bind(y,\varepsilon[|5|]_\mu, bind(x,\varepsilon[|3+y|]_\mu,\rho)$

$YH^0 = bind(y,\perp_{Den}, bind(x,\perp_{Den},\mu)$

...

Exercise 7 (pts 4)

(a) Rephrase in Ocaml the following integer product, written in Haskell, that is non-strict on the second argument:

```
h = \ n m -> if (n=0) then 0
                 else if (n=1) then m else n*m
```

(b) Write and comment the application, in Ocaml, of the rephrased h to 1 and (f 5) when f is defined by:

```
let rec f = fun n -> f n
```

Exercise 8 (pts 4)

(a) Provide a Ocaml definition for the finitely approximated, infinite list, oddN, of all the odd natural numbers;

(b) Use oddN to provide a definition of function that, given n, returns the sum of the first n odd natural numbers.

Exercise 9 (pts 3) Put into the C expression below, the abstract syntax operators Val and Den:

```
A[*v+j] = x = y + A[x+1]
```