CSE305 **Sample Prelim II** **Spring 2023**

Closed book, no electronics, one notes sheet allowed but otherwise closed notes, closed neighbors, 75 minutes after 5-minute read-in period. There are **four** problems, with points totaling 100 and subdivided as shown.

Do all answers **only on these sheets**. You may use the front and back of this exam for extra scratch space. *Show your work*, and explain your reasoning where it is naturally called for—doing so may help for partial credit.

(1) (33 pts. total)

The following C# and C++ programs are supposed to be identical, when the call-by-reference markers (`ref` in C#, `&` in C++) and the call to `Foo()/foo()` are respectively commented in or out.

```
using System;

public class Global {
  int x;

  public Global() { x = 0; }

  public int B(/*ref*/ int z) {
    int y = x + z;
    z = 3;
    return y;
  }

  int Foo() { return 0; }

  public void Test() {
    int x = 7;
    int y = 2*x + /*Foo()*/ + B(/*ref*/ x);
    Console.WriteLine("x={0}, y={1}",x,y);
  }

  public static void Main(string[] args){
    (new Global()).Test();
  }
}
```

```
using namespace std;
#include <stdio.h>
class Global {
  int x;
public:
  Global(): x(0) { }

  int bb(int/*&*/ z) {
    int y = x + z;
    z = 3;
    return y;
  }

  int foo() { return 0; }

  void test() {
    int x = 7;
    int y = 2*x + /*foo()*/ + bb(x);
    printf("x=%d, y=%d\n",x,y);
  }
};

int main() {
  (new Global())->test();
}
```

(a) Diagram the stack frames for the calls to `Test/test` and `B/bb`, showing the storage objects created inside the frames and changes to them, when call-by-value is used (as written). Since the programs are supposed to be identical, one diagram suffices for both languages. You may ignore the frames for `Main/main` and the `Global()` constructor, and may treat the *class* `Global` as just an external destination for static links (you need not show the links this time). What values of `x` and `y` are printed? (12 + 3 = 15 pts.)

(b) If dynamic scoping were used, what would the values be? (3 pts.)

(c) Now redo the stack-frame sketch when `ref/&` is commented in, so that call-by-reference is used. Now what values of `x` and `y` are printed? (Actually, there are two different possible answers for `y`, see below!) (6 + 3 = 9 pts.)

(d) Weirdly, when `int&` is used in the C++ code, the call `foo()` is commented-in, and the code is compiled using `g++` (with or without optimization), the program prints a different answer for `y`! Why might this happen? Explain the two ways in which `y` might be computed. (6 pts. You do not need to show detail at the level of our "rudimentary stack-language")

(2) (15 + 6 = 21 pts.)

Execute the following program in our "rudimentary stack model." Show the contents of the stack after each operation, and give the final contents of the variables when the computation ends.

```
w   4   x   3   5   -   store   +   x   fetch   *   store   pop
```

Then say which of the following C/C++/Java/Javascript assignment statements the Postfix came from. Draw the "C expression tree" for the correct one.

(a) `w = 4 + (x = 3 - 5)*x;`

(b) `w = (4 + (x = 3 - 5))*x;`

(c) `w = (4 + (x = 5 - 3))*x;`

(d) `w = 4 + (x = 3) - 5*x;`

(3) (4+4+4+4+3+3 = 22 pts.)

True/False: Please write out the words `true` or `false` in full. No justifications are needed—though if you want to write something like "`BS`" instead, please explain...

(a) If `foo` is a function that always returns 0 but the expression `exp + foo()` can have a different value from `exp` itself, then the programming language is not referentially transparent.

(b) Replacing a `switch(exp)` statement in C++ by an `if (exp == label1) { ...} else if (exp == label2) { ...} else if ...` construct that tests the expression against each label value in turn, always produces code that behaves the same way as the `switch` statement.

(c) In a Java for-loop `for(int i = 0; i <= 20; i++) {...}`, if `i = 15` after the loop exits, then the loop terminated early.

(d) Every `for`-loop in a Java program can be translated into an equivalent `foreach` loop.

(e) In a typical imperative language such as C, Java, or Ada, a case of a function calling itself recursively without terminating would always overflow the system stack.

(f) The case in (e) could also overflow the system heap, if the function created extra linked-list nodes.

(4) (21 + 3 = 24 pts.)

(a) Write in OCaml a function `tabulate (f,n)` that returns the list of pairs `[(0,f(0)), (1,f(1)), ..., (n,f(n))]`. Your `tabulate` must consist of one call to a tail-recursive helper function `th`, which you can nest inside `tabulate` or leave at top level (your choice).

(b) Looking at your code for (a), can it be applied to a function f with integer arguments and any return type, or only for functions f that return `int`s?

End of Exam

Extra Practice Problem. There are reasons this is similar to a static-scoping question, so it would be "kind-of fair." It also connects to method *co-/contra-variance* to come in my Tue. 4/26 lecture, which will *not* be on Prelim II.

(5) (3+9+9 = 21 pts.)

Consider the following object hierarchy, written in C#. (*Notes:* Whether methods are `public` or `private` does not matter for this question—my leaving everything as C# "internal" is fine. The ellipses `...` indicate that the code of the constructor or method is immaterial to the question. In C#, `+` can be used to concatenate a number to a String.)

```
using System;

class Shape {
   internal Shape(...) {...}
   internal /*virtual*/ float Perimeter() {...}
   internal /*virtual*/ void Stats() {
      Console.WriteLine("Perimeter is: " + Perimeter());
   }
}

class Ellipse : Shape {
   internal Ellipse(...) {...}
   internal /*virtual*/ float Area() {...}
   internal /*override*/ float Perimeter() {...}
   internal /*override*/ void Stats() {
      Console.WriteLine("Perimeter is: " + Perimeter());
      Console.WriteLine("Area is: " + Area());
   }
}
```

```
class Circle : Ellipse {
   internal Circle(...) {...}
   internal /*override*/ float Area() {...}
   internal /*override*/ float Perimeter() {...}
}

class Polygon : Shape {
   internal Polygon(...) {...}
   internal /*override*/ float Perimeter() {...}
}

class Rectangle : Polygon {
   internal Rectangle(...) {...}
   internal /*override*/ float Area() {...}
   internal /*override*/ void Stats() {
      Console.WriteLine("Perimeter is: " + Perimeter());
      Console.WriteLine("Area is: " + Area());
   }
}

class Square : Rectangle {
   internal Square(...) {...}
   internal /*override*/ float Perimeter() {...}
}
```

Also suppose that we have the declarations:

```
Circle c = new Circle(...);
Ellipse e = new Ellipse(...);
Polygon p = new Polygon(...);
Rectangle r = new Rectangle(...);
Square s = new Square(...);
```

Note that there is no Shape object.

(a) Which of those objects satisfy "(___ `instanceof Polygon`)" in Java-terms? (3 pts.)

(b) Make a 5×3 grid with rows labeled `c,e,p,r,s` and columns labeled `Area, Perimeter, Stats`. In each box, indicate the class whose method gets called in a statement of the form `row.column()`, or write `Illegal` if it is an illegal call. For instance, in row `r`, column `Stats`, which method gets called by `S.stats()`? (Don't pay attention yet to the fact that `Stats()` itself calls other methods... just say what gets called first.)

(c) Same question as (b) when `virtual` and `override` are commented-in.