CptS355 - Programming Language Design
Spring 2020

Sakire Arslan Ay, PhD

School of Electrical Engineering and Computer Science
Washington State University

Final Sample
Solutions

Time Limit: 75 minutes

This sample exam is illustrative of a final exam in this course. DO NOT USE IT AS YOUR EXCLUSIVE STUDY GUIDE. Review the lecture notes, quizzes as well as assigned textbook and other reading, your projects and the class calendar. You should expect about 40% of the exam to be on material since the previous exam with the rest split between the materials that was on the first two exams.

Exam terms:

- This is an individual assignment. You should work on the exam problems yourself and provide your own solutions. You should not talk or communicate with anybody during the exam. You can’t share or receive solutions through any means including email, messaging, and online communication.
- You are allowed to use ONLY the class notes as reference. You should not search the solution of the problems on the Internet or anywhere else.
- If you do not comply with the above rules, you will receive 0 for this exam and you will be reported to WSU Office of Community Standards.

Final exam is on Wednesday May 6 @ 4:00pm. The exam paper will be posted on Blackboard.

1) (**16 pts**) Short answers and multiple choice.

a) In Haskell what does `[1]:[2]:[3]:[]` evaluate to?

`[[1],[2],[3]]`

b) Consider the following Haskell function:

```
my_function c x = c
```

What will the following evaluate to?

```
y = my_function True
```

a) `True`
b) `False`
c) Will return a function
d) Will give an error

c) Suppose that in a C++ program class A has 2 virtual methods, and that 1000 instances of class A have been allocated. How many v-tables are there for class A?

Only one.

d) C++ uses _________________ for virtual methods.
a) Dynamic scoping
b) Dynamic typing
c) Dynamic dispatch
d) None of the above

2a) (6 pts) Recall that Haskell uses static scoping. What value will be bound to **x** when the following Haskell code is evaluated? Explain how you determined your answer.

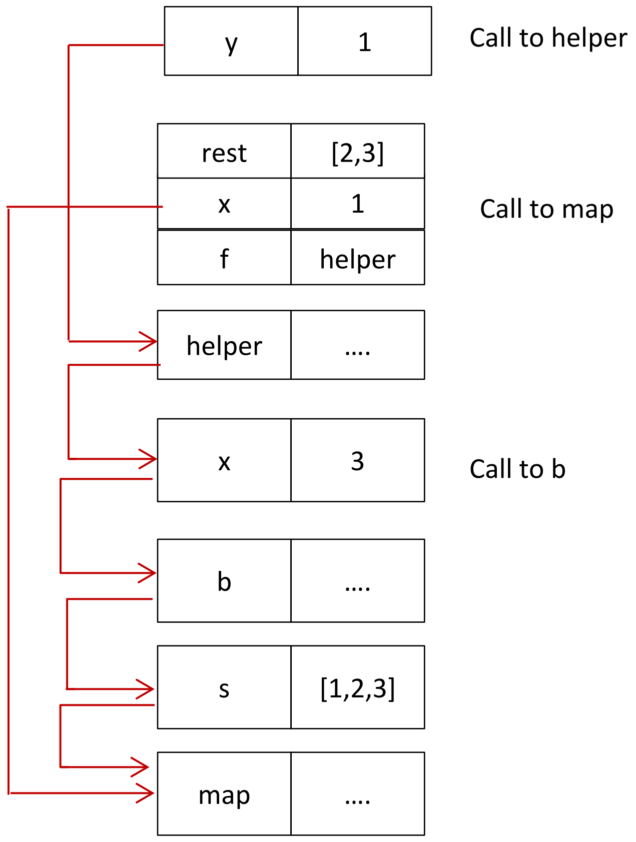
```
map f [] = []
map f (x:xs) = (f x):(map f xs)

s = [1, 2, 3]
b x = let
        helper y = x + y
      in
        map helper s

x = b 3
```

The below shows the stack after function helper is called in map.
Assuming static scoping, the ML code will return `[4,5,6].`

2b) (8 pts) *What if* Haskell used dynamic scoping. What would be the value of the expression given in 2a) in that case? Explain how you determined your answer.



`helper` function is called within `map.` `In helper`, if dynamic scoping is used, Haskell will retrieve the most recent declaration of `x` , which is the `x` in `map`. In `map`, `x` holds the first value in the input list which is also passed to `helper` as the `y` argument. So the above code will evaluate to [2,4,6] when using dynamic scoping.

3) Haskell

```
ok2 _  _  [] = []
ok2 f c (x:xs) = (f c x):(ok2 f c xs)
```

a) (5 pts) YES or NO: is the function ok2 tail recursive? Explain.

No. It is not tail recursive. After the recursive call returns, it ‘cons’ the result of (f c x) to the value returned from the recursive call. Since it needs to do additional work after recursive call returns it can’t utilize the same stack frame for each recursive call, therefore it is not tail recursive.

b) (5 pts) Give an example expression that correctly uses the ok2 function above.

```
add a b  = a + b

ok2 add 10 [1,2,3]
```

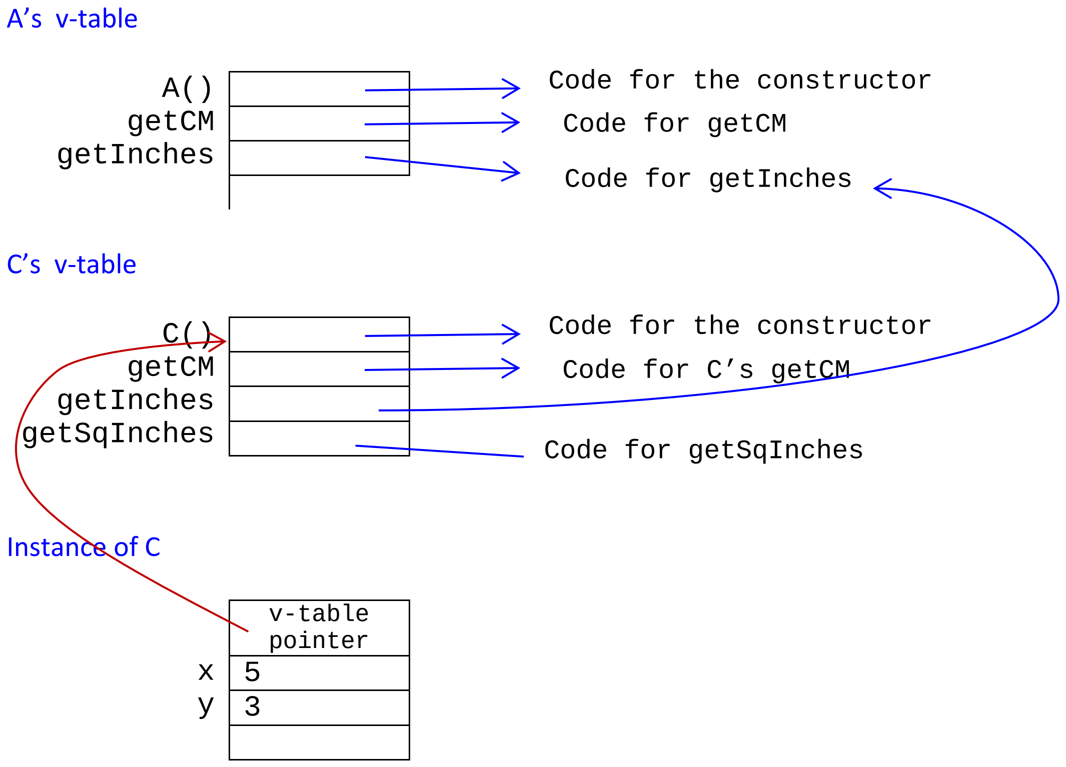
returns

```
[11,12,13]
```

4) (12 pts) Consider the Java code below. (Java methods are all virtual in the sense of "virtual methods" in C++).

```
class A {
   int x;
   A() {x = 5;}
   int getCM() {return x;}
   double getInches() { return (x*0.4); }
};
class C extends A {
   int y;
   C() { super(); y = 3; }
   int getCM() { return (x*y); }
   double getSqInches () { return (x*y*0.16); }
};
class M {
 public static void main(String argv[]) {
    A a = new C();
    C c = (C) a; // cast a to type C
    {*}
 }
}
```

4a) (7 pts) Draw the object layout in memory for an instance of class C, showing the data members and the v-table.



4b) (5 pts) What is the value of `a.getCM()` at the point marked `{*}`?
What is the value of `c.getCM()` at the point marked `{*}`?
What is the value of `c.getInches()` at the point marked `{*}`?

`a.getCM()` will return `15`
`c.getCM()` will return `15`
`c.getInches()` will return `2.0`

5) Consider the Haskell code:

```
x = 6
addx y = y + x  -- **HERE**
map f []     = []  -- this is just the standard map: no tricks
map f (x:xs) = (f x) : (map f xs)
z = map addx [1]
```

5a) (5pts) Knowing that Haskell uses static scoping, what value will be bound to **z** at the end of this code? Explain your answer.

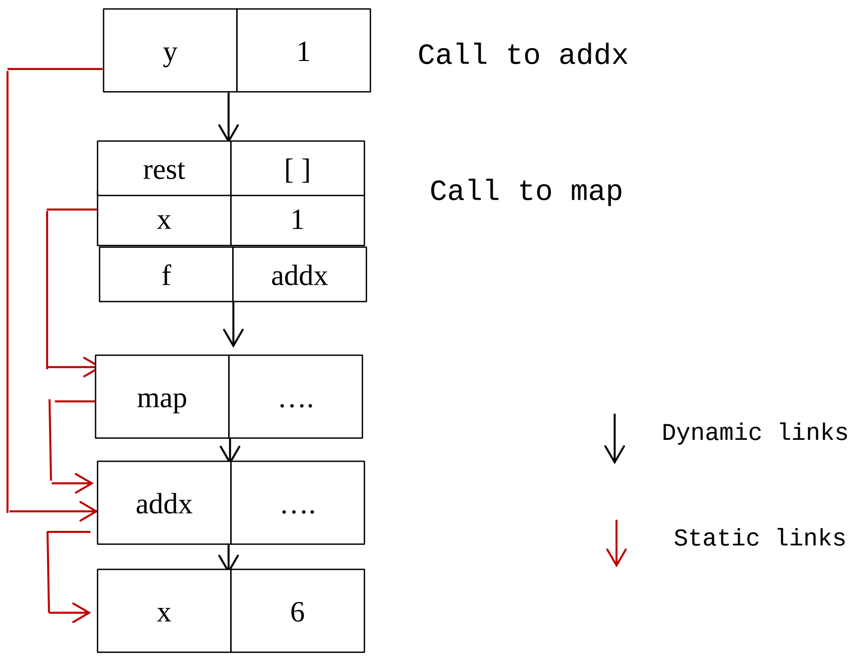
`[7]`
According to static scoping rules when the body of function `addx` is evaluated, the x value is 6. Therefore `(map addx [1])` will add 6 to each element in the input list `[1]`.

5b) (10 pts) Draw the stack of activation records (ARs) as it exists at the point labeled **`**HERE**`** in the code above. The intention is that you draw a stack of ARs that shows the definitions of `x, addx, map` and `z` as well as the call from the last line to map and the call from map to `addx`.)

In your drawing clearly identify in each AR:
- the dynamic link
- the static link
- the names of any variables held in that AR, and
- the values of those variables.

For ARs that correspond to function calls, label the AR with the function name and arguments. For ARs that contain function definitions, show the code body of the function as "…".

See the next page.



6) (6 pts) Consider the Python code:

```
x = [0, 1, 2, 3]
def g(a, b):
    a[2] = False
    b = 7
g(x, x)
print (x)
```

Remembering that Python uses the pass-by-value calling convention, and that list values are represented as references, what is printed by the program?

```
[0, 1, False, 3]
```

7) (6 pts) Consider the following example pseudo-code language:

```
integer j;
function P(integer k)
{
     print(k);
     j = j+1;
     print(k);
}
j = 10;
P(2*j);
```

7a) (3 pts) What is the output when k is passed by value?

```
20
20
```

7c) (3 pts) What is the output when k is passed by name?

```
20
22
```

8) (10pts) What does the following Java program print?

```java
import java.awt.Point;

class Test {
   public static void reset1(Point q) {
       q.x = 0;
       q.y = 0;
   }
   public static void reset2(Point q) {
       q = new Point(0,0);
   }

   public static void main(String[] args) {
       Point p = new Point(10, 20);
       System.out.println("before reset1: p.x = " + p.x);
       reset1 (p);
       System.out.println("after reset1: p.x = " + p.x);

       p = new Point(10, 20);
       System.out.println("before reset2: p.x = " + p.x);
       reset2 (p);
       System.out.println("after reset2: p.x = " + p.x);
   }
}
```

```
before reset1: p.x = 10
after reset1: p.x = 0
before reset2: p.x = 10
after reset2: p.x = 10
```

9) Multiple Inheritance

(11 pts) Multiple inheritance poses a number of problems for language designers and implementers. Describe those problems. Illustrate your answer with example code.

A class C may inherit from two base classes A and B that both define a method for a message M or both define an instance variable v.
When we access C.M, it will be ambiguous whether A's M method or B's M method is invoked. Also should two copies of v be inherited, or one. This is called a name clash.

For example, assume we define the following:

```
class Rectangle extends Shape { float area() { ... } }
class Rhombus extends Shape { float area() { ... } }
class Square extends Rectangle, Rhombus {}

Square sq = new Square();
```

When we call `sq.area(),` it will be ambiguous whether Rectangle’s or Rhombus’s area method is called.