

Homework 4: Procedure and Data Abstraction(25pts)

Due Oct 26, 2011

1. For the following Pascal program, what is the value of x after the call to *apply* finishes? Draw the activation records that are on the runtime stack just prior to the return from the first call of procedure *foo*. Include all entries in the ARs. Use line numbers for return addresses. Clearly label the values of local variables and parameters. Label each AR with its procedure name.

```
1: program main(input,output)
2: var x : int;
3: procedure foo(y : int)
4:   begin   x := x * y   end
5: procedure add(x : int)
6:   begin foo(x+1) end
7: procedure apply(procedure f(a : int))
8:   var x : int; var y : int;
9:   begin
10:    x := 5; y := x-1;
11:    f(x);
12:    foo(y)
13:   end
14: begin
15:   x = 3;
16:   apply(add)
17: end
```

2. Suppose we have the following C++ code.

```
class A
{
  protected:
    int a,b,c;
  public:
    A(int x) { a = x; b = x * 2; c = b * 2; }
    virtual void shift(int m) { a = a + m; b = b + m; c = c + m; }
};

class B : public A
{
```

```

private:
    float y;
public:
    B(int _x, float _y) : A(_x), y(_y) {}
    float get_value() { return y; }
    virtual void shift(int m) { A::shift(m); y = y * m; }
    virtual void apply() { y = y + c; }
};

class C : public B
{
private:
    int z;
public:
    C(int _x, float _y) : B(_x,_y) { z = b + a; }
    void shift() { B::shift(z); }
    virtual void move(int m) { z = z + m; }
};

```

Draw a picture of the memory layout for the classes A, B, and C.