Assignment 1

Due: 11:59PM EST, Nov 2 2017

All questions must either be answered individually, or in a team of at most 3 members.

1. [12 points] For the following C program:

(a) **[4 points]** Draw the stack activation records after the call to r on line 13 (i.e. the execution reaches line 7)
(b) **[4 points]** Draw the stack activation records after the call to p on line 14 (i.e. the execution reaches line 2)
(c) **[4 points]** Describe where variables r and x are located on lines 4 and 5 and how they are accessed during the execution of p

(When you draw the activation records, show the control links, and fill in the local names in each activation record)

```
(1)   int x;
(2)   void p(void)
(3)   { double r = 2;
(4)     printf("%g\n",r);
(5)     printf("%d\n",x);
(6)   }
(7)   void r(void)
(8)   { x=1;
(9)     p();
(10)  }
(11)   void q(void)
(12)  {  double x = 3;
(13)     r();
(14)     p();
(15)  }
(16)  main()
(17)  { p();
(18)    q();
(19)    return 0;
(20)  }
```

2. **[15 points]** Read the following Java code:

```
class Point {
   protected int x;
```

```
    protected int y;
    public Point (int x, int y) { this.x = x; this.y = y; }
}

class ColorPoint extends Point  {
   protected int color;
   public ColorPoint(int x, int y, int color)  { super(x,y); this.color = color; }
}

public class test {
    public static void main (String[] args){
        ColorPoint pa = new ColorPoint(1,2,3);      //Line 0
        Point pb = new Point(4,5);
        // pa = pb;                    // Line 1
        // pb = pa;                    // Line 2
        // pa = (ColorPoint)pb;        // Line 3
        // pb = (Point)pa;             // Line 4

    }
}
```

(a) **[3 points]** After Line 0 is executed, does the heap create one continuous area with **x**, **y**, and **color** fields inside, or does the heap create two distinct areas, one with **x**, **y** and the other with **colo**r?
(b) **[3 points]** Suppose Line 1 is included in the program but Lines 2, 3, 4 are commented out, should the program compile? If not, why? If yes, what is the value of field x of pa?
(c) **[3 points]** Suppose Line 2 is included in the program but Lines 1, 3, 4 are commented out, should the program compile? If not, why? If yes, what is the value of field x of pb?
(d) **[3 points]** Suppose Line 3 is included in the program but Lines 1, 2, 4 are commented out, should the program compile? If not, why? If yes, what would be run-time behavior of the program?
(e) **[3 points]** Suppose Line 4 is included in the program but Lines 1, 2, 3 are commented out, should the program compile? If not, why? If yes, what would be run-time behavior of the program?

3. [15 points] Given the following C++ declarations, draw the VMT of each class and the layout of memory for a dynamically allocated object of **each class**.

```
class A {
  public: int a;
          virtual void g();
          void h();
};
class B public A {
  public: virtual void h();
          void f();
          int b;
```

```
};
class C  public B {
  public: void h();
          void p();
}
```

4. [8 points] Many people think the Java "interface" mechanism is an encapsulation mechanism. Their arguments are as follows. Suppose we have an interface A, being implemented by class B. If programmers consistently use type A every time an instance of B is referred to, then any method, say m, defined in B but not specified in A is hidden and cannot be accessed by any code holding a reference of type A. There however is a fatal flaw in this argument. Explain how a piece of code referring to an instance of B (with type A) can in fact access m.

5. [20 points] Write a Haskell function **osublist lt** that computes all sublists of a list **lt** that have an odd size (i.e., size of 1, 3, etc).
e.g. >sublist [1,2,3]
[[1],[2],[3], [1,2,3]]

The order of elements in the sublist does not matter. For example, the answer [[1,2,3], [1], [2],[3]] is also correct.

6. [10 points] Write a Haskell function **keepl k lt** that keeps every **k**th element of a list **lt**.
e.g. > keepl 2 [3,4,5,6,7,8,9]
[4,6,8]

7. [20 points] Read paper "Quick Introduction to Type Systems," at https://www.cs.princeton.edu/courses/archive/spring12/cos320/typing.pdf

and answer questions:

(a) **[2 points]** Why does a typing judgment require a type environment sometimes?
(b) **[2 points]** What does it mean to say a program is well-typed?
(c) **[5 points]** On top of page 2, there is a rule labeled (2) "Function Applications." Consider the following Haskell program:
(\x -> x +1) 3
explain what this rule means for typechecking this program.

(d) **[5 points]** Write a derivation for the following program (where + is used in the prefix form):
+ (+3 1) 2
(e) **[3 points]** What does decidability mean for a type system?
(f) **[3 points]** Modern compilers routinely throw out "type errors" for programs that fail to compile. What is a type error?

Submission Instructions:

- Write down all answers other than Haskell programming in a standard text file, such as txt, doc, or pdf, and name it assignment1.suffix where suffix is one of the above.

- Write down the answers to the Haskell programming questions in a file named assignment1.hs.

- Write a README file (text file, do not submit a .doc file) which contains

    - You name(s) and email address(es). PLEASE list your team members if any.

    - Whether your Haskell programs were tested on department machines.

    - Briefly describe anything special about your submission that the TA should take note of.

- 4. Place all files under one directory with a unique name (such as [bmail_id]_1 for assignment 1, e.g. davidL_1).

- 5. Tar the contents of this directory using the following command.
  tar –cvf [directory_name].tar [directory_name]
  e.g. tar -cvf davidL_1.tar davidL_1/

- 6. Upload your tared file on mycourses.

Each team only needs to submit one copy. It does not matter which member on the team submits.