CMSC 330 Exam #2 practice questions #1 Fall 2012

Do not open this exam until you are told. Read these instructions:

1. This is a closed book exam. **No notes or other aids are allowed.** If you have a question during the exam, please raise your hand. Each question's point value is next to its number.
2. **You must turn in your exam immediately when time is called at the end.**
3. In order to be eligible for as much partial credit as possible, show all of your work for each problem, and **clearly indicate** your answers. Credit **cannot** be given for illegible answers.
4. You will **lose credit** if ***any*** information above is incorrect or missing, or your name is missing from any side of any page.
5. Parts of this page and of two other pages are for scratch work. If you need extra scratch paper after you have filled these areas up, please raise your hand. Scratch paper must be turned in with your exam, with your name and ID number written on it. Scratch paper **will not** be graded.
6. To avoid distracting others, **no one** may leave until the exam is over.
7. The Campus Senate has adopted a policy asking students to include the following handwritten statement on each examination and assignment in every course: "*I pledge on my honor that I have not given or received any unauthorized assistance on this examination.*" Therefore, **just before turning in your exam**, you are requested to write this pledge **in full** and **sign it** below:

_____

_____

Good luck!

1. Suppose you dropped and broke your cell phone, so you don't have your contact list or phone list any more. However, you do have the OCaml interpreter downloaded onto your computer. After you have finished weeping over your broken phone, write an OCaml function that can be used to create a contact list, for later lookup:

   - Your function should be called `create_contact_list` and it should have one parameter, a list of strings. The list consists of a sequence of strings, alternating between your friends' first names (in no particular order) and their phone numbers.
   - Your function `create_contact_list` should **create and return a function** that can then be used to look up your friends' phone numbers, given their first names. For instance:

```
let lookup_friends = create_contact_list ["Tommy"; "(301) 123-4567";
                                          "Tammy"; "(301) 111-2222";
                                          "Derek"; "(301) 987-6543";
                                          "Varun"; "(301) 222-3333"];;

lookup_friends "Derek";;
- : string = "(301) 987-6543"
lookup_friends "Tommy";;
- : string = "(301) 123-4567"
```

   - You may assume the list passed into `create_contact_list` will be valid and consist of alternating names and phone numbers as described. You may assume the function it returns will only be called with names that are in the list, and that your friends all have unique first names. (You can always get rid of friends who have the same first names as others.)

   It doesn't matter if your function would cause incomplete match warnings. To receive full credit, your solution **should not use any references**.

2. Write a context–free grammar that generates the following language:

$$\left\{ a^{\mathrm{m}} b^{\mathrm{n}} \;\middle|\; \mathrm{m} + \mathrm{n} \text{ is } \textbf{not} \text{ a multiple of } 3 \right\}$$

For full credit, your grammar should be **unambiguous**. In order that your answer can be graded as accurately and quickly as possible, please use consecutive nonterminals beginning with S (S, T, U, etc.) when writing your grammar.

SCRATCH WORK AREA
(this area will not be graded)

3. Consider the following program, written in C synax:

```
#include <stdio.h>

int x= 1;

int f(int a, int b) {
  int c= a + x;
  x += 1;
  x += b + 2;
  return a + c;
}

int main() {
  int y= 7;
  y= f(x + 1, y + x);
  printf("%d %d\n", x, y);
  return 0;
}
```

What would the program print if C used call–by–name as its default parameter transmission method? _____

SCRATCH WORK AREA
(this area will not be graded)

4. Java, subtyping, and generics.

   a. Consider the following Java program:

```
class A {
  void g() {
    System.out.println("g1");
  }
}

class B extends A {
  void g() {
    System.out.println("g2");
  }
}

class C {
  static void f(A a) {
    System.out.println("f1");
  }

  static void f(B b) {
    System.out.println("f2");
  }

  public static void main(String[] args) {
    A x= new A();
    A y= new B();
    B z= new B();

    f(x);
    f(y);
    f(z);

    x.g();
    y.g();
    z.g();
  }
}
```

   Give the program's complete output below:

   _____

   _____

   _____

   _____

   _____

   _____

   _____

   b. Suppose the following method is added to class `C`:

```
static <T extends B> void h(T t) {
  System.out.println("h");
}
```

   Which, if any, of `x`, `y`, and `z` could be passed into `h`? _____

   c. What kind of polymorphism are Java's generics an example of?

   _____