COMS W4115
Programming Languages and Translators
Homework Assignment 3

Prof. Stephen A. Edwards
Columbia University

Due December 2nd, 2015
at 11:59 PM

Please write your name clearly on the paper.

Do this assignment alone. You may consult the instructor and the TAs, but not other students.

1. For the following C array,

```
int a[2][3];
```

   assume you are working with a 32-bit little-endian processor with the usual alignment rules (e.g., a Pentium).

   (a) Show how its elements are laid out in memory.

   (b) Write an expression for the (byte) address of `a[i][j]` in terms of a, i, and j.

   (c) Verify parts a) and b) by writing a small C program that allows you to **test your hypothesis**. Examine the assembly language output with the C compiler's -S flag (e.g., `gcc -O -S array.c`). Such a program should be simple and contain and access such an array, but not be so simple that the compiler can optimize most of it away. **Turn in an annotated assembly listing** that explains how it verifies your hypothesis. Make sure the assembly listing is no more than about 40 lines, either by simplifying your program or trimming the output.

2. For a 32-bit little-endian processor with the usual alignment rules, show the **memory layout** and **size in bytes** of the following three C variables.

```
union {
  short a;  /* 16-bit */
  struct {
    int b;  /* 32-bit */
    char c; /* 8-bit */
  } s;
} u1;
```

```
struct {
  short a;
  char b;
  short c;
  int d;
} s1;
```

```
struct {
  int d;
  short a;
  short c;
  char b;
} s2;
```

3. Draw the layout of the stack just before *bar* is called in *foo*. Indicate storage for function arguments, local variables, return addresses, and stored frame pointers. Indicate where the stack and frame pointers point.

```
void bar(int x, int y);

void foo(int a, int b)
{
  int c, d, e;
  bar(1, 2);
}
```

4. Draw the layouts of a Circle and a Rectangle object as well as the virtual tables for their classes. Indicate how the runtime decides to call the appropriate *area* function for *s1* in *main*.

```
public class Shape {
   public double area() { ... }
}

class Circle extends Shape {
   private double diameter;
   public double area() { ... }
}

class Rectangle extends Shape {
   private double height, width;
   public double area() { ... }
}

public class Main {
  public static void main() {
    Shape s1 = new Rectangle(35, 42);
    System.out.println( s1.area() );
  }
}
```