1. Given the following Java code:

```java
class C {
    public int foo(C p) { return 1; }
}
class D extends C {
    public int foo(C p) { return 2; }
    public int foo(D p) { return 3; }
}
C p = new C();
C q = new D();
D r = new D();
int i = p.foo(r);
int j = q.foo(q);
int k = q.foo(r);
```

(Remember that in Java every object is accessed through a pointer) Which methods get called in the three calls, i.e., what are the values of i, j, and k? Explain how the method selection works to result in these values?

2. Show the run-time stack with all activation record instances, including static and dynamic chains, when execution reaches position 1 in the following skeletal program (in C-style syntax). Assume that the programming language is statically scoped but does not allow functions as return values.

```c
void Foo () {
    void B (int);

    void A () {
        void C () {
            ... // position 1
        } // end of C()
        ...
        C ();
    } // end of A()

    void B (int flag) {
        void D () {
            ...
            B (1);
        } // end of D()
        if (flag)
            A ();
        else
            D ();
        ...
    } // end of B()
    B (0);
    ...
}
```

The calling sequence for this program for execution to reach C is: main calls Foo, Foo calls B, B calls D, D calls B, B calls A, and A calls C.
3. Define the terms *static* and *dynamic scope* and explain the difference between them. If possible, present an example in C, C++, or pseudo-code, where the code behaves differently under static scooping and dynamic scooping.

4. Below code is written in Java defining two classes OBJ1 and OBJ2 and an interface OBJ0. Mark the lines of code of class OBJ2 with invalid assignments and calls and explain them.

```java
interface OBJ0{
    static int var0 = 5;
    public int call3();
}
class OBJ1 implements OBJ0{
    static int var1; public int var2;
    protected double var3;
    private double var4;
    void call2(){...}
    public static void call4(){...}
}
class OBJ2 extends OBJ1
void call2(){...}
public int call3(){...}

OBJ1 obj=new OBJ1();
obj.call1();
obj.call2();
obj.call3();
obj.call4();
OBJ1.call4();
obj.var1  = 1;
OBJ1.var1 = 2;
obj.var2  = 3;
OBJ1.var2 = 4;
var1 = 5;
var2 = 6;
var3 = 7.0;
var4 = 8.0;
var2 = OBJ0.var0;
var2 = OBJ1.var0;
var2 = OBJ2.var0;
var2 = var0;
var1 = OBJ1.var0;
var1 = obj.var0;
call1();
call2();
call3();
```

4. Consider the following skeletal C program.

```c
void fun1(){
    int b, c, d, e, f, h;     /* definition def1 */
...
while (...) {
    int c, d, e, f, k;     /* definition def2 */
    while (...) {
        int a, b, c, g, j;     /* definition def3 */
        ...
    }
} // end of while()
```
Given that main calls fun1; fun1 calls fun2 and assuming that dynamic scoping is used for subprogram calls, what variables are visible during execution of the line marked as HERE? Include with each visible variable, the name of the definition in which it was defined.