1. Virtual Methods (5 + 5 + 5 + 5 = 20 points)

Consider the following Java program:

```java
class C0 {
    int a() { return 10; }
    int b() { return 20; }
}
class C1 extends C0 {
    int a() { return 11; }
    int cee() { return 31; }
}
class C2 extends C1 {
    int a() { return 12; }
    int b() { return 22; }
}
class Main {
    public static void main(String[] args) {
        C0[] x = new C0[3];
        x[0] = new C0(); x[1] = new C1(); x[2] = new C2();
        for (int i=0; i<x.length; i++)
            System.out.println(x[i].a()); System.out.println(x[i].b());
    }
}
```

1a. (5 points) What is the vtable of class C0?
1b. (5 points) What is the vtable of class C1?
1c. (5 points) What is the vtable of class C2?
1d. (5 points) What does the program print?

2. Exceptions (10 + 10 = 20 points)

2a. (10 points) What is a `finally` clause in Java? What is it good for?
2b. (10 points) What happens in Java when a `catch` clause throws an exception? Give an example!

3. Type Inference (5 + 5 + 10 = 20 points)

Consider the following OCaml function:

```ocaml
let rec f = function [] -> [] | h::t -> (2 * h)::(f t);
```

3a. (5 points) Describe in words what `f` does.
3b. (5 points) What is the type of `f`?
3c. (10 points) Show the type inference steps for `f`. 
4. Concurrency (10 + 5 + 5 = 20 points)
Consider the following (buggy) bounded buffer get method:

```java
public String get() {
    while (count == 0) { /* do nothing */ }
    count--;
    out++;
    return buf[out % buf.length];
}
```

Assume that `buf.length` is 3, and that the buffer is already initialized with two strings "s1" and "s2". Assume that each line of code in the `get` method is atomic.

4a. (10 points) Show an interleaving of two calls to `get` with an undesirable race condition. In other words, there are two consumer threads, both of which are executing `get` at the same time. Your answer should be a table that starts like this:

<table>
<thead>
<tr>
<th>thread</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>while(count == 0) {}</td>
</tr>
</tbody>
</table>

4b. (5 points) Fix the code to prevent the race condition.

4c. (5 points) Fix the code from (b) to prevent deadlock.

5. Parameter Passing (5 + 5 + 5 + 5 = 20 points)
Consider the following Ada program:

```ada
with Ada.Text_IO;
procedure Main is
  X : Integer;
  procedure P(Y : in out Integer) is begin Y := 1; X := 2; end P;
begin
  X := 0;
  P(X);
  Ada.Text_IO.Put_Line(Integer'Image(X));
end Main;
```

5a. (5 points) What does the program print if “in out” means call-by-value?

5b. (5 points) What does the program print if “in out” means call-by-reference?

5c. (5 points) What does the program print if “in out” means call-by-value-result?

5d. (5 points) Which parameter passing mode does Ada use for “in out” Integer parameters?