

## Homework Assignment #5

1. C++ uses a termination model of exception-handling, which raises the question of whether or not stack frames are unwound while searching for a handler. Write C++ programs to determine
  - (a) how many times the exception object gets copied in a long distance `throw`.
  - (b) what happens to the stack when a `throw` is not caught (look at a core-dump).

Hand in code with an explanation of what it shows.

2. Suppose that we have a calling environment where
  - `d` is a double variable
  - `a` is an array of `real` numbers;
  - `a[3]` is 3.3;
  - `a[4]` is 4.4;
  - `i` is an integer variable with value 3;
  - `fred` is a globally defined function, which in C++ syntax is defined to be

---

```
double fred(double x, int j)
{
    int i ;
    i = j ;
    j = j+1 ;
    x = x * j ;
    return 2 * x ;
}
```

---

- and we execute `d = fred(a[i], i)`; Explain carefully what happens with
  - (a) call-by-value;
  - (b) call-by-reference;
  - (c) call-by-name;
  - (d) call-by-textual-substitution; and

(e) call-by-copy-return.

3. Explain carefully how the environment in which the body of a non-static member function is executed is created in C++. That is, what bindings come from the calling environment, what bindings come from the environment associated with the class of the member function, and what bindings come from the object calling the member function.

This assignment is due in class at the beginning of class 2007-11-27, or you may e-mail it to Robert Pringe (<pringler@unbc.ca>).