

# CSCI-1200 Computer Science II — Spring 2006

## Lab 8 — Templates & String Operations

### Checkpoint 1

Here is a simple templated function that takes 3 arguments, a boolean and two arguments of the type T (which will be specified at runtime). If the first argument is true, the function returns the second argument, otherwise it returns the third argument.

```
template <class T>
T select(bool flag, T a, T b) {
    if (flag)
        return a;
    else
        return b;
}
```

Type this into your programming environment and try these examples:

```
cout << select(true, 5, 7) << endl;
cout << select(false, "Sally", "Fred") << endl;
```

Write a new templated function that takes in an integer  $n$  and a vector of the templated type and returns the element in the vector at index  $n$ . Test your program with a variety of inputs.

**To complete this checkpoint:** Show a TA your tested and debugged function.

### Checkpoint 2

Here is the `Even & Odd` code from the test:

```
bool Odd(int number);

bool Even(int number) {
    if (!Odd(number)) return true;
    else return false;
}

bool Odd(int number) {
    if (!Even(number)) return true;
    else return false;
}
```

Type it into your programming environment and try it out. Use the debugger to see what goes wrong. Rewrite the code, but you are only allowed to use subtraction and testing if something is equal to zero. Also `Odd` cannot directly call itself and `Even` cannot directly call itself.

**To complete this checkpoint:** Show a TA what goes wrong in the original version and how you fixed the problem.

### Checkpoint 3

After Checkpoint 2, get your test from your TA. Type your answers from Problem 3 into your development environment and create a simple `main()` function with a while loop that reads (from `cin`) a string and then a number (the substring length) and then calls `SubStringSort`. Use this to test and debug your code. Try a variety of different inputs and see if you can break or cause unexpected behavior. How robust is your solution?

**To complete this checkpoint:** Show a TA your tested and debugged program.