Lecture Notes 6

Template Specialization

- Template Specialization – Template redeclaration where all parameters are specified
  - Example:
    ```
    template <typename T>
    int Compare(const T &l, const T &r){
        if(l < r){
            return -1;
        }
        if(l > r){
            return 1;
        }
        return 0;
    }

    template <>
    int Compare(const char *const &l, const char *const &r){
        return strcmp(l,r);
    }
    ```
  - Built in specialization – `std::vector<bool>`

C vs. C++

- Capabilities missing in C
  - No `classes`, only `structs`
  - Structs cannot have member functions, only data members
    - Example C++
      ```
      struct StructName{
          int D1;
          int foo(int val) const;
      };
      int StructName::foo(int val) const{
          return D1 + val;
      }
      ```
    - Example C
      ```
      struct StructName{
          int D1;
      };
      int StructName_foo(const struct StructName *st, int val){
          return st->D1 + val;
      }
      ```
  - No access control – Everything is `public`, no `private` or `protected`

- No inheritance – Must create new type with duplicating members
- No templates – Generic programming can somewhat be done through macros
  - Example C++
    ```
    template <typename T>
    T min(const T &left, const T &right){
        return left < right ? left : right;
    }
    ```
  - Example C
    ```
    #define min(left,right) (left < right) ? left : right
    ```
- No pass by reference, only pass by value (use pointers for emulating pass by value)
  - Example C++
    ```
    void foo(int &param){
        param *= param;
    }
    ```
  - Example C
    ```
    void foo(int *param){
        *param *= *param;
    }
    ```
- No `new` or `delete`, need to use `malloc` and `free`
  - Example C++
    ```
    int *Ptr = new int[11];
    ...
    delete [] Ptr;
    ```
  - Example C
    ```
    int *Ptr = malloc(sizeof(int) * 11);
    ...
    free(Ptr);
    ```
- No `iostreams`, need to use `printf`, `scanf` and derivatives
  - Example C++
    ```
    int I = 15;
    double D = 3.3;
    std::cout<<I<<" "<<D<<std::endl;
    ```
  - Example C
    ```
    int I = 15;
    double D = 3.3;
    printf("%d %lf\n",I,D);
    ```
- No `std::string`, need to use `char` strings
  - Example C++
    ```
    std::string Str = "Hello World";
    ```

- Example C

```
char Str[] = "Hello World";
```