ECE326 – Fall 2019: Week 5 Exercise Questions

1. True or False [1 mark each]

Circle T is true, otherwise circle F for false.

1. Generic programming is a subset of metaprogramming. **T** **F**

2. If no deep copy is required (e.g. class has no pointer), move semantics performs no better than copy semantics. **T** **F**

3. If template specialization is not used (i.e. not instantiated), its code is not generated for the final executable. **T** **F**

4. For template `T foo()`, you can write `int a = foo()` to instantiate the function template `foo` with an `int` parameter. **T** **F**

5. The new operator in C++ couples heap allocation and constructor invocation. **T** **F**

2. Short Answers

1. Use container_of to return a pointer to the parent object of member field base. **[2 marks]**

```
struct base {
  int x, y, z;
};

struct derived {
    int a;
    struct base b;
    char c[10];
};

struct derived * get_derived(struct base * b) {









}
```

2. Implement binary search algorithm using a function template, assume swap template function has already been implemented and the array is sorted. **[5 marks]**

```
template<typename T>  /* find index of val in array of size n */
int binary_search(const T & val, T * array, int n) {









}
```

3. Implement a template class named Triple that is a tuple of 3 elements of the same type. Overload enough operators so that binary search template you implemented above can be instantiated for Triple. Use lexicographical order. **[5 marks]**

```
template<typename T>
class Triple {









};
```

3. Generic Programming **[10 marks]**

Create a generic Queue class without using templates. Implement the Queue using a singly linked list, with the member functions, `push_back`, that pushes new elements to end of the queue, `front`, which returns the first element of the queue, and `pop_front`, which removes the first element of the queue.

4. Template Programming **[10 marks]**

Using the generic Queue made in Question 3, write a FIFO class template, which allows type-safe use of the generic Queue class for any parameterized type. Use move semantics for `push_back` instead of copy semantics.