Experience and skill working with C++:

- Teach me C++ in 5 minutes.
- How do you detect memory leaks and prevent the memory from leaking?
- Do you use STL? (Standard Template Library). What are the advantages? Describe some examples.
- What is an abstract base class and how do you create one in C++?
- What is the difference between a template and a macro?
- What does pass by reference mean? Why use that instead of pass by ptr? When should use one over the other?
- What is the difference between an object, reference and pointer in C++?
- In C++ what is function overloading and function overriding. Give examples.
- Declare a class Vechicle and make it an abstract data type.
- What is the v-ptr?
- What is the difference between const int * pOne and int *const pTwo?
- What is an exception in C++? How does it work? When is it used?
- What is a "virtual" method/function and what are they used for?
- Explain the following code –

```
Class Base
{
  public:
      Base();
      Base(Base const & rhs);
      virtual ~Base();
  private:
      int *pI;
}

class Derive: public Base
{
  public:
      Deriv();
      Deriv(Deriv const& rhs);
      virtual ~Deriv();
  private:
      int *pI;
}
int main()
{
  Base *pD = new Deriv();
  delete pD;
}
```

C++ Programming

1. What is a class?   How does it differ from a struct?

2. What is a constructor/destructor?

3. Can a constructor be protected or private? (yes) Why would you want to do this? (to control creation of new instances of this class)

4. What is the "default constructor"? (one with no arguments)

5. What is a virtual function? What is it used for? (method that must be defined in a subclass; an object containing a virtual method cannot be instantiated)

6. Explain the difference between overloading & overriding.

7. What is operator overloading? (Allows programmer to define how an operator like '+' is handled by this class. Ex: + may be used as a concatenator.)

8. What are exceptions? How are they used in a program?

9. Is it ok not to catch an exception in a method? Why?

10. Describe the difference between static and dynamic binding. (static takes place at compile time; dynamic happens at run time)

11. What is recursion? When is it useful?