**Q.** In C++, which class methods are automatically generated by the compiler, under what circumstances, and what is their access protection and signature? Explain why you may or may not want to provide your own implementations of these methods.

**Ans**. In C++, special member functions are the methods which are automatically generated by the compiler. They will only be generated if they are used in the program but are not declared explicitly by the program.
These special member functions are

- Default constructor: Constructors can be very useful for setting initial values for certain class' member variables.
- Copy constructor: It initialize one object from another of the same type. It copies the object and pass it as an argument to a function. If it is not defined, the compiler itself creates one.
- Move Constructor: Move constructors typically "steal" the resources held by the argument, rather than making copies of them, and leave the argument in some valid, otherwise indeterminate state.
- Copy assignment operator: We typically declare a copy assignment operator when copy-and-swap idiom cannot be used.
- Move assignment operator: It is used for transferring a temporary object to an existing object. The move assignment operator can be overloaded. The move assignment operator is different than a move constructor because a move assignment operator is called on an existing object, while a move constructor is called on an object created by the operation.
- Destructor: A destructor is a special member function of a class that is executed whenever an object of its class goes out of scope or whenever the delete expression is applied to a pointer to the object of that class.

I would most likely want to implement these methods, especially constructors when my class requires it. Apparently, there is no generic answer to this solution. It all depends on the type of class and what it contains. Most of the time one do not need to define anything. Ideally one do not need to define anything other than a regular constructor. So, I would definitely make use of constructors if my class requires it. Similarly, If I want some code to run on destruction and free the memory, I would define a destructor. If the implicitly generated special member functions do what you intend, then none of them should be declared explicitly. If you have an explicit definition for any of (destructor, copy/move constructor/assignment), then as a rule of thumb, you need a definition for each of them.