Answer the following questions. Each question is worth 5 points.

Differentiate high level and low level programming languages  
High level programming is programming with code that’s readable by the programmer without having to fully understand how a particular hardware was designed but it needs a compiler for the machine to understand (e.g. C, C++, C#, Java and etc.) meanwhile, Low level programming is where only the machines themselves can understand, it will also not need a compiler to interpret messages and has direct access to the central processing unit.

1. Differentiate structured and object oriented programming?  
   Structured programming follows a step by step process of a particular program that mainly focuses on the process itself and the procedures of the system. While OOP uses the concept of inheritance, methods, and attributes that can be passed down. Think of it as a family tree where it can expand indefinitely and may still inherit ancestral properties.
2. Differentiate class and object.  
   A class is a blueprint from which you can create the instance, i.e., objects, used to bind data as well as methods together as a single unit, have logical existence, doesn't take any memory spaces when a programmer creates one, and the class has to be declared only once. While an object is the instance of the class, which helps programmers to use variables and methods from inside the class, acts like a variable of the class, have a physical existence, takes memory when a programmer creates one, and objects can be declared several times depending on the requirement.
3. What is a source code?  
   A source code is code that’s written by a programmer using a high level programming language often written in text editors built for programming, it could be a script, an executable and etc.
4. What is a byte code?  
   A byte code is a machine readable code that is converted from a human readable code which contains binary digits from English words and special characters by an interpreter that is usually stored in a class file.
5. What do you mean by architecturally neutral?  
   When Java is compiled, it is not compiled into platform specific machine, rather into platform-independent byte code. When you write a piece of Java code in a particular platform (e.g. Windows, Linux, Mac) and generated an executable code .class file. You can execute a .class file on any system but the only condition is that the target system should have Java Runtime Environment installed in it. In short, Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.