* Research and write about the differences between Abstract Classes and Interfaces.
* Research and write about why Unit Testing is important and what benefits it adds.

There are a few differences between Abstract classes and Interfaces. Both have their own importance and different uses. The Abstract class can have non-abstract and abstract methods, while Interface’s can have only abstract methods. While using Abstract class, you can’t have multiple inheritances, while interfaces allow multiple inheritances. The interface has only static and final variables. The Abstract class can use final, non-final, static, and non-static variables. The Abstract class can provide implementation of interface but the interface can not provide implementation of abstract class. They both have different keywords both key words will declare what they are. Abstract class can implement many different Java interfaces and extend another java class, but the interface and only extend Java interface. If you wanted to extend an Abstract class you would use keyword extends. Interface can be implemented using keyword implements. Java interface is public by default and the Abstract class can have class members like private, public, and protected. As described you would use both for different things and both have their own importance.

Unit Testing is defiantly important and it makes coding not only better and easier to read but the quality of the code is greatly improved. Unit testing makes the process more flexible. This helps because when more and more code is written, some of the older code will often need to be changed, this can be difficult because it can cause code to not run properly. If you have unit tests in place you won’t have to worry about messing anything up. Unit testing will identify every defect before the code is sent for integration testing. It will also find issues with software early before the wrong code impacts future code. Unit testing allows programmers to refactor code at a later date and make sure the module still works correctly and also reduces bugs when changing the existing functionality. Unit testing will also add documentation within in the code so other programmers can understand what is happening. It simplifies the debugging process. If you write the test first, it forces the coder to think through the design and what the code needs to do before the code is written. This will keep the coder focused and help create better designs. Lastly Unit testing can reduce the cost of fixing bugs that are found in the future because it helps you fix problems as they occur. Unit testing is very important and needs to be utilized if user wants to design quality code.