**Purpose**

The purpose of this unit was to introduce us to Desk Checks, and how to test our software to ensure that we get the desired output.

**Concepts**

**Desk Check** - A physical run through of a programs logic to ensure that the desired output is achieved. This is often used in software testing to find bugs and to ensure that the solution to a problem is valid.

**Software Testing** – Implemented in a variety of ways, software testing is meant to reduce bugs and undefined behavior in programs by making sure that a functions output meets pre-defined expectations.

**Naming Conventions** – In programming using defined naming conventions makes it easier to follow a programs execution and helps teams of programmers work together since they have a common set of variable names to pull from and they are designed in the same way. For example, if a programmer uses camel case naming for one variable, they should do their best to use camel case for all variables.

**Mathematical Operators** – Mathematical operators in programming allow for complex algorithms that can do more than simply print a string onto the screen. These can range from simple arithmetic to higher level mathematics, and in the case of game development, linear algebra and vector manipulation.

**Float** – A float is a piece of numerical data with a decimal. In PHP a floats maximum value of 1.7976931348623E+308 with a maximum precision up to 14 digits. The precision gained from using floats over integers allow for more complex equations and higher accuracy in calculations.

**Implications**

The purpose of this unit was to introduce us to critical analysis of a program. As programmers become build larger and more complex programming interfaces, they often run into the problem of making sure that their code actual does what they believe they have programmed it to do. It becomes necessary at that point to develop testing methods meant to pass or fail the execution of their program based on a pre-defined expected output compared against the actual output of the program or functions within the program.

The desk check is one example of such tests and in a more professional environment, entire departments are dedicated to designing tests and implementing those tests, often in continuous deployment, meaning the tests are performed on every single version of their software before that version can be released, and it is done on a regular basis.

Unfortunately, these unit tests are only valuable if the person writing the tests really understands the implications of the test. Their value starts to degrade the more the tests are generalized. For that reason, unit tests and things like desk checks are often best performed on singular functions rather than entire programs at once.

The lack of reasonable and accurate unit testing has led to a shift in programming recently and the old trend of functional programming has started to see a resurgence in popularity. I for one welcome our new functional overlords.