**WEEK 2 REPORT**

On Tuesday 16/06/15;

**2.1 TEAM REPORT**

The facilitator, Mr. Kawalya asked us to talk about our teams, elaborating on our strategies and plans. Each team was composed of a maximum of four(4) members and there were a total of 3 teams in the shift. My team for which i am the team leader is called ***corevelvet*;** our emphasis as a team was to approach all work with a keen eye and an emphasis on detailed presentation.

**2.2 GIT AND GITHUB**

I was introduced to git; a distributed version control system that's designed to handle all kinds of projects, automate deployment and aid continuous integration.

I also learnt the steps taken in creating a git repository in a project folder and the commands necessary to add the project’s files and folders in the newly created repository.**(Appendix A)**.

I was admonished to use git for all my internship and future projects

I was also introduced to github; which is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management functionality of Git as well as adding its own features.Unlike Git, which is strictly a command-line tool, GitHub provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every projects.

In addition I learnt the steps **(Appendix A)**necessary to add a local repository to github.

A local repository is the one created on your machine, whereas the the github repository is the remote one.

**2.3 RUBY PROGRAMMING LANGUAGE**

I was introduced to the ruby language, as it is the language that i would predominantly use in my internship projects. I was also taught how to quickly get started with any programming language by;

* Knowing how to setup the environment
* Learning the syntax and semantics

I was instructed on the various ways of installing ruby under different operating systems;

- In Windows; Using the ruby installer <rubyinstaller.org>

- In Linux; using package managers, RVM: ruby version manger which can mange multiple versions of ruby

After installing ruby, i learnt how to write ruby programs(Appendix B) and also run them(Appendix B).

I was also introduced to **irb** which is short for interactive ruby**. Irb** can accessed from the command line to test out syntax and statements before incorporating them in a ruby program.

**2.4 RESEARCH ASSIGNMENTS**

I was instructed to read about the following;

i) Bootstrap affix: The affix plugin allows a <div> to become affixed to a location on the page. You can also toggle it's pinning on and off using this plugin.

ii) ssh-keys: SSH keys serve as a means of identifying yourself(i.e your PC) to an SSH server using public-key cryptography and challenge-response authentication

**On Thursday 18/06/15**

**2.5 WEB DEVELOPMENT**

I was introduced to developing web application development. In this I learnt about bootstrap which is front-end framework for faster and easier web development. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

I also was introduced to a number of components i would use;

- ruby on rails as the framework I was to use in developing the back-end of the applications

- SQlite as the database for development and MySQL for deployment

The instructor continued elaborating on ruby as i learnt about strings, numbers and file input/output operations in the language**(Appendix B)**. I was also introduced to loops and how to use them in the language. One application of strings told us was that for security reasons, strings that make up data sent over a network are reversed.

I was given an assignment of writing a basic calculator in ruby.

My various internship code is at <https://github.com/phalbert/calc>

**On Saturday 20/06/15**

I continued in my study of ruby when i was introduced to variables and the different scopes in which they are used in the language and these include;

- Instance Variables - These exist within the context of an instance. Each instance of a class will have its own specific set of these variables, independent from other instances of the same class.

- Class Variables - These exist at the class level, and are accessible from the class scope (inside the class, but outside of any methods) and within class methods.

- Global Variables - These are rarely used as they break the rules of encapsulation. However, Ruby supports them and they are used in a few places.

Next, i learnt of constructors whose purpose is to initiate the state of an object. The constructor in Ruby is called **initialize.(Appendix B)**

I also learnt about inheritance in ruby and how its different from the implementation in java; polymorphism; and arrays.

I learnt that every data type in Ruby has a **class**: a set of defined properties and functionality.

APPENDIX A

**Creating a git repository for a project requires running the following commands**;

git init // initializes a repository in the folder

git add file // stages a file to be committed to initialized repository

git commit -m “description” //adds the staged file to the repository

**Adding a local repository to github;**

- Create a new repository on github in your account

- Then add the local repository using the following commands;

git remote add origin “url\_of\_remote\_repository”

git push -u origin master

APPENDIX B

**Strings** are a sequence of characters denoted by single or double quotes. Here are some examples of strings:

* "a"
* "puts"
* "John's book"
* "12+100"
* 'To be or not to be, that is the question...'

You can add strings together – this is sometimes referred to as **concatenation**. You can't subtract or multiply strings together:

puts "abc" + "def" #=> "abcdef"

**Numbers**

There are two types of numbers :

Fixnum: that is an integer like 42

Float: A decimal number like 3.8

**File input and output;**

The File class supplies the basic methods to manipulate files. The following script opens a new text file in "write" mode and then writes "Hello file!" to it:

fname = "sample.txt" # variable fname
somefile = File.open(fname, "w")
somefile.puts "Hello file!"
somefile.close

"w" stands for **write mode. Such other modes include “a” for append, “r” for read**

Constructor example;

irb> class Fruit #class name

| def initialize # constructor

| @kind = "apple"

| @condition = "ripe"

| end

| end

nil

irb> f4 = Fruit.new # arguments to new are delivered to initialize.

"a ripe apple"