Readme File for VM Execution

For the milestone one, I have used modular component design approach to make different classes for each component of the Virtual Machine such as CPU, Memory, Register, Loader, and many others. This would further help when the project grows, and more granular components of the operating system are added to it. The main class has VMShell which starts when the program starts running. The key classes of my implemented Virtual Machine are Loader, CPU, Main Class, Register, and Memory.

CPU has functions which help in Fetch-Decode-Execute cycle from the Main memory.

Loader class helps in loading the program from the .osx binary file.

Memory class simulates the main memory as a single-dimensional array.

Register class manages registers including the general purpose and special purpose registers.

VMShell helps in command line interaction of the user with the VM.

USAGE

1. Run the MainClass.py file.
2. To start with you can type shell which creates a new shell and then load a program using load filename -v.
3. Then you can run the program using run filename -v.
4. To see the memory and the data into the register you can type coredump filename -v which outputs the whole memory and register values into an external file.
5. Then you can type errordump filename which shows all the error with their timestamp.
6. The second program does not load at the same location if some program is already loaded ready to be executed.
7. You can type exit, quit or ctrl+c to exit the current shell.