

```

//
// Equation.swift
// Kintematics Calculator
//
// Created by Luke Deratzou on 6/6/18.
// Copyright © 2018 Luke Deratzou. All rights reserved.
//

import Foundation

class Equation {
    private var listOfVars: [PhysicsVariable]
    private var equationName: String
    private var answer: [PhysicsVariable] = [PhysicsVariable]()
    //Add the setUpUnits function to this init
    init(listOfVars: [PhysicsVariable], equationName: String) {
        self.listOfVars = listOfVars
        self.equationName = equationName
    }
    //Delete this init
    init(listOfVars: [PhysicsVariable], equationName: String,
        isUnitsEnabled: Bool) {
        self.listOfVars = listOfVars
        self.equationName = equationName
        if isUnitsEnabled {
            setUpUnits()
        }
    }

}

private func setUpUnits() {
    for i in listOfVars {
        if i.unit != i.getSIUnits() {
            //i.value = Helper.CONVERT_UNITS(physicsVar: i, toSI:
            true)
            i.value = Helper.CONVERT_UNITS(from: i.unit, to:
            i.getSIUnits(), value: i.value)
        }
    }
    //for loop going through each var in the PhysicsVariable list
    //then checks if the var.units equals the var's SI units (have
    it as a getter func)
    //If it does not equal, call on a convert function from
    UnitConverter ViewController and set equal to new value
}

func doEquation() {

```

```

switch equationName {
case "kinematics":
    let temp = KinematicsEquations.init(listOfKnowns:
        listOfVars)
    temp.doKinemtaicsEquation()
    for i in 0...1 {
        answer.append(temp.getAnswers()[i])
    }
case "forces":
    let temp = ForceEquations.init(listOfVars: listOfVars)
    temp.doEquation()
    answer.append(temp.getUnknown())
case "kinetic energy":
    let temp = KineticEnergyEquations.init(listOfKnowns:
        listOfVars)
    temp.solve()
    answer.append(temp.getUnknown())
case "gravitational force":
    let temp = GravitationalForceEquation.init(listOfVars:
        listOfVars)
    temp.solve()
    answer.append(temp.getAnswer())
default:
    print("error w/ doEquation")
}
}

func getAnswer() -> [PhysicsVariable] {
    return answer
}

func getListOfVars() -> [PhysicsVariable] {
    return listOfVars
}
}
}

```