16.35
Pset 1

1) Real time systems and software
- Better understand how to implement sensor-reading code into robotic systems. This could be used with drones and other systems used in space exploration and sample collecting.
- Be able to write autonomous programs for robotic systems, which can be used for unmanned robots that are sent into space.
- Learn how to implement parallel controls into a robot. Again can be useful for having robotic systems that need to do multiple concurrent functions in space.

2) Spent about 20 minutes on problem 2
- What is the version of Python installed on athena?
  They have version 2.7. Found from Athena Cluster.

- How do you define a constructor in Python?
  To define a constructor in Python, you must use the __init__ method.
  http://www.diveintopython.net/object_oriented_framework/defining_classes.html - d0e11896

- How do you raise exceptions in Python?
  To raise exceptions in Python, you have to use a try statement. Basically the user or program can try some operation, and if it is invalid, the program will go to the except clause of the try statement.
  https://docs.python.org/3/tutorial/errors.html

- When using the Python unit testing framework, how can you check if a particular exception is raised during execution?
  You would use the assertRaises() method.
  https://docs.python.org/2/library/unittest.html

- How do you define a "main" function in Python?
  To define a "main" function, you would have to type def main():
  http://www.guru99.com/learn-python-main-function-with-examples-understand-main.html

4) Requirements and Unit testing
Methods that should have unit testing

- Control Constructor
- GroundVehicle Constructor
- setPosition()
- setVelocity()
- controlVehicle()
- updateState()

I believe that these methods should be unit tested because the yare actually performing some sort of algorithm, which has a chance of failing.

Methods that don't need unit testing

- getSpeed()
- getRotVel()
- getPosition()
- getVelocity()

I don't think these methods need unit testing because they are simply returning a value. They are not changing anything. (Don't have an input)

Input equivalent classes
Control(s, omega)

- s<5
- 5<=s<=10
- s>10
- omega<-pi/4
- -pi/4<=omega<=pi/4
- omega>pi/4

GoundVehicle(pose, s, omega)

- x<0
- 0<=x<=100
- x>100
- y<0
- 0<=y<=100
- y>100
- theta<-pi
- -pi<=theta<=pi
- theta>pi
- s<5
- 5<=s<=10
- s>10
- omega<-pi/4
- -pi/4<=omega<=pi/4
- omega>pi/4

setPosition(pose)

- x<0
- 0<=x<=100
- x>100
- y<0
- 0<=y<=100
- y>100
- theta<-pi
- -pi<=theta<=pi
- theta>pi

setVeloctiy(vel)

- s<5
- 5<=s<=10
- s>10
- omega<-pi/4
- -pi/4<=omega<=pi/4
- omega>pi/4

controlVehicle(c)

- s<5
- 5<=s<=10
- s>10
- omega<-pi/4
- -pi/4<=omega<=pi/4
- omega>pi/4

updateState(sec, msec)