"""

Student: Robin G. Blaine

Date: October 17, 2017

Class: \_Python Programming

Assignment (Module 1, Chapter 2, Project 5):

An object's momentum is its mass multiplied times its velocity.

Write a program that accepts an object's mass (in kilograms) and

velocity (in meters per second) as inputs and then outputs its

momentum.

Pseudocode:

Input mass ("Enter the object's mass in kilograms: ")

Input velocity ("Enter the objects velocity in meters per second: ")

momentum = mass \* velocity

Output mass ("The object's momentum is", mass, "kg m/s")

"""

mass = float(input("Enter the object's mass in kilograms: "))

velocity = float(input("Enter the object's velocity in meters per second: "))

momentum = mass \* velocity

print("The object's momentum is", momentum, "kg m/s")

"""

Student: Robin G. Blaine

Date: October 17, 2017

Class: \_Python Programming

Assignment (Module 1, Chapter 2, Project 6):

The kinetic energy of a moving object is given by the formula

KE = (1/2)mv^2, where m is the object's mass and v is its velocity.

Modify the program you created in Project 5 so that it prints the

object's kinetic energy as well as its momentum.

Pseudocode:

Input mass ("Enter the object's mass in kilograms: ")

Input velocity ("Enter the objects velocity in meters per second: ")

momentum = mass \* velocity

kineticEnergy = .5 \* mass \* velocity^2

Output mass ("The object's momentum is", mass, "kg m/s")

Output kineticEnergy ("The object's kinetic energy is", kineticEnergy, "kg m^2/s^2")

"""

mass = float(input("Enter the object's mass in kilograms: "))

velocity = float(input("Enter the object's velocity in meters per second: "))

momentum = mass \* velocity

kineticEnergy = .5 \* mass \* velocity \*\* 2

print("The object's momentum is", momentum, "kg m/s")

print("The object's kinetic energy is", kineticEnergy, "kg m^2/s^2")

"""

Student: Robin G. Blaine

Date: October 17, 2017

Class: \_Python Programming

Assignment (Module 1, Chapter 2, Project 7):

Write a program that calculates and prints the number of minutes in a year.

Pseudocode:

minutesPerStandardYear = 365 \* 24 \* 60

minutesPerLeapYear = 366 \* 24 \* 60

minutesPerAverageYear = 365.25 \* 24 \* 60

"""

minutesPerStandardYear = 365 \* 24 \* 60

minutesPerLeapYear = 366 \* 24 \* 60

minutesPerAverageYear = 365.25 \* 24 \* 60

print("The number of minutes per standard year: ", minutesPerStandardYear)

print("The number of minutes per leap year: ", minutesPerLeapYear)

print("The number of minutes per 'average' year:", minutesPerAverageYear)

"""

Student: Robin G. Blaine

Date: October 17, 2017

Class: \_Python Programming

Assignment (Module 1, Chapter 2, Project 8):

Light travels at 3 \* 10^8 meters per second. A light-year is the distance a

light beam travels in one year. Write a program that calculates and

displays the value of a light-year.

Pseudocode:

lightSpeed = 3 \* 10 \*\* 8

secondsPerYear = 365 \* 24 \* 60 \* 60

lightYear = lightSpeed \* secondsPerYear

Output lightYear ("A light year is", lightYear, "meters")

"""

lightSpeed = 3 \* 10 \*\* 8 # light speed in meters per second

secondsPerYear = 365 \* 24 \* 60 \* 60 # days \* hours \* minutes \* seconds

lightYear = lightSpeed \* secondsPerYear

print("A light year is", lightYear, "meters")

"""

Student: Robin G. Blaine

Date: October 17, 2017

Class: \_Python Programming

Assignment (Module 1, Chapter 2, Project 9):

Write a program that takes as input a number of kilometers and prints the

corresponding number of nautical miles. Use the following:

\* A kilometer represents 1/10,000 of the distance between the North Pole

and the equator.

\* There are 90 degrees, containing 60 minutes of arc each, between the

North Pole and the equator.

\* A nautical mile is 1 minute of an arc.

Pseudocode:

Input kilometers ("Enter the number of kilometers: ")

nauticalMiles = kilometers \* .54

Output nauticalMiles (kilometers, "km =", nauticalMiles, "nautical miles")

"""

kilometers = float(input("Enter the number of kilometers: "))

nauticalMiles = kilometers \* .54

print(kilometers, "km =", nauticalMiles, "nautical miles")