SFWRTECH 3PR3:

Procedural and Objective Oriented Programming Concepts

(Assignment #2)

Student Name: Dojae Kim

Student Number: 400420323

Professor Name: Dr. Seshasai Srinivasan

**Objective**

The purpose of this Assignment 2:

1. To give students understanding python structure.

2. To write, test, and debug simple python programs.

3. To implement, and execute conditional statement iteratively until a given condition is satisfied.

4. To understand how to import module from python library.

**Introduction**

This report presents a design of a collision test with the equation below:

**s = max(0, 𝑣𝑡 + 0.5𝑎𝑡 2)**

**Input Specification**

|  |  |
| --- | --- |
| **Input Variables** | **Limit** |
| Distance (d) | [5, 10] |
| Initial Velocity (v) | [1, 10] |
| Acceleration (a) | [-100, 0] |
| Time (t) | A positive number less than 10 |

**Output Specification**

|  |  |
| --- | --- |
| **Conditional Statement** | **Result** |
| Distance (d) = Displacement (s) | The object will collide |
| Distance (d) < Displacement (s) | The object will collide |
| Distance (d) > Displacement (s) | The object will not collide |

**Source Code**

**Part1**

# Student name: Dojae Kim  
# Student number: 400420323  
# Student email: kim408@mcmaster.ca  
# Lecture: SFWRTECH 3PR3  
# Assignment 2 Part 1  
  
import math  
  
distance = float(input('Enter your distance [5 - 10]: '))  
while distance < 5 or distance > 10:  
 distance = float(input('Please enter distance [5 - 10]: '))  
  
initial\_velocity = float(input('Enter initial velocity [1 - 10]: '))  
while initial\_velocity < 1 or initial\_velocity > 10:  
 initial\_velocity = float(input('Please enter initial velocity [1 - 10]: '))  
  
acceleration = float(input('Enter acceleration [-100 - 0]: '))  
while acceleration < -100 or acceleration > 0:  
 acceleration = float(input('Please enter acceleration [-100 - 0]: '))  
  
travel\_time = float(input('Enter travel time less than 10: '))  
while travel\_time >= 10:  
 travel\_time = float(input('Please enter travel time less than 10: '))  
  
displacement = (initial\_velocity \* travel\_time) + (0.5 \* acceleration) \* (math.pow(travel\_time, 2))  
  
print('\n====================================================')  
  
if displacement >= distance:  
 print('\tThe object will collide')  
 print('====================================================')  
 print('Distance \t\t\t Displacement')  
 print(distance, 'm', '\t\t\t\t', displacement, 'm')  
  
  
else:  
 print('\tThe object will not collide')  
 print('====================================================')  
 print('Distance \t\t\t Displacement')  
 print(distance, 'm', '\t\t\t\t', displacement, 'm')  
print('====================================================')

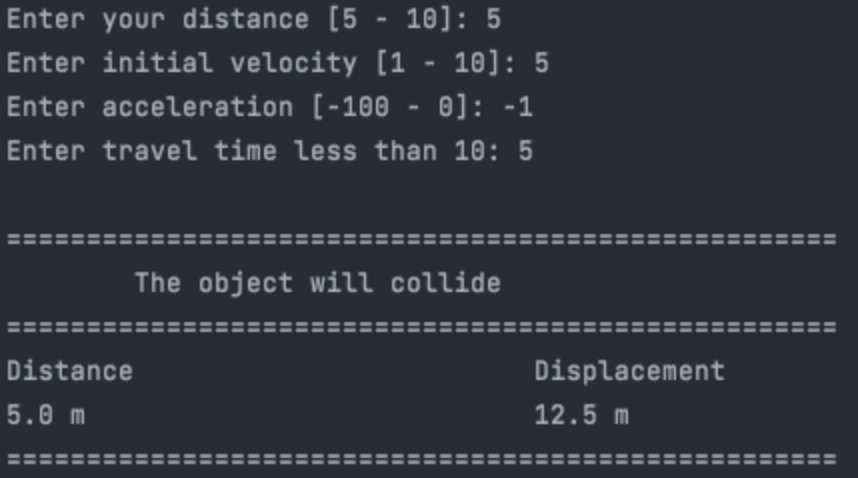
**Part 1 Sample Output**

**Sample 1:**

(Input: Distance: 5 m, Initial velocity: 5 m/s, Acceleration: -1 m/s2, Time: 5 s)

(Output: Displacement 12.5 m (**Distance** < **Displacement**)

**Result** -> “The object will collide”

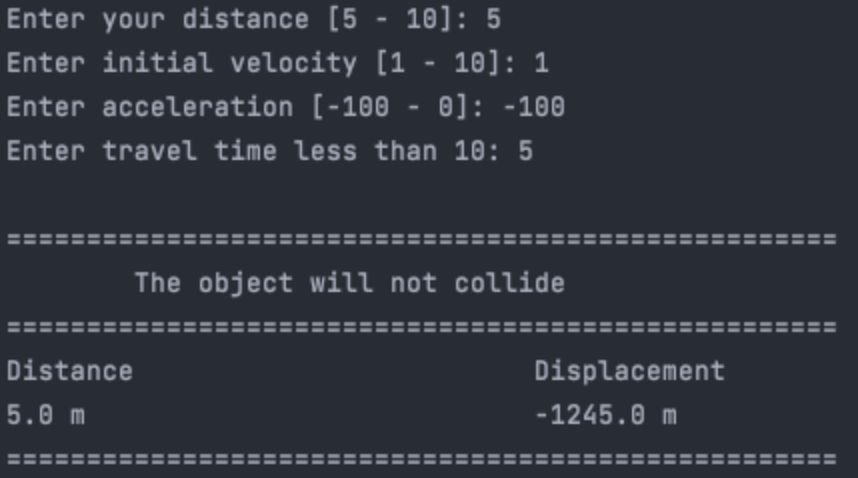


**Sample 2:**

(Input: Distance: 5 m, Initial velocity: 1 m/s, Acceleration: -100 m/s2, Time: 5 s)

(Output: Displacement -1245.0 m (**Distance** > **Displacement**)

**Result** -> “The object will not collide”



**Part2 Source Code**

# Student name: Dojae Kim  
# Student number: 400420323  
# Student email: kim408@mcmaster.ca  
# Lecture: SFWRTECH 3PR3  
# Assignment 2 Part 2  
  
import math  
import numpy  
  
displacement = 0  
  
distance = float(input('\nPlease enter distance [5 - 10]: '))  
while distance < 5 or distance > 10:  
 distance = float(input('\nDistance is out of bounds, please enter distance [5 - 10]: '))  
  
initial\_velocity = float(input('\nPlease enter initial velocity [1 - 10]: '))  
while initial\_velocity < 1 or initial\_velocity > 10:  
 initial\_velocity = float(input('\nVelocity is out of bounds, please enter initial velocity [1 - 10]: '))  
  
print('\n===============================================================================================================')  
print('Distance', '\t', 'Displacement', '\t\t', 'Acceleration', '\t\t', 'Travel Time', '\t\t\t', 'Description')  
print('===============================================================================================================')  
while True:  
 for acc in numpy.arange(-50.0, 0.0, 0.2):  
 for time in numpy.arange(0.0, 10.0, 0.1):  
 displacement = (round(initial\_velocity, 2) \* round(time, 2)) + (0.5 \* round(acc, 2)) \* (round(math.pow(time, 2), 2))  
 round(displacement, 2)  
 if displacement < distance:  
 print(round(distance, 2), 'm', '\t\t', round(displacement, 2), 'm', '\t\t', round(acc, 2), 'm/s^2', '\t\t', round(time, 2), '\t\t', 'Object A will not hit object B')  
 elif displacement >= distance:  
 print(round(distance, 2), 'm', '\t\t', round(displacement, 2), 'm', '\t\t', round(acc, 2), 'm/s^2', '\t\t', round(time, 2), '\t\t', 'Object A will hit object B')  
 print('===============================================================================================================')  
 break

**Part 2 Sample Output**

Sample 1:

(Input : Distance: 5 m, Initial velocity: 5 m/s)

\* Acceleration and Travel Time value will automatically generate based on conditional statement

1. Acceleration : -50 to 0.0 by 0.2 increment

2. Travel Time : 0 to 9.9 by 0.1 increment

