CSCI 1015 – Programming Assignment 5
Classes and Methods

Learning Outcomes

- Define a class that simulates a real world object.
- Write fields and methods for a class.
- Write a program that runs a simulation using objects.

Required Reading

Savitch - Sections 5.1, 5.2 (pp. 301-313)

Instructions

For this assignment you are going to write a program that simulates what happens when a ball is thrown at a particular angle and initial velocity.

You will need to create a class called `Ball` that keeps track of the status of the ball at a given time, and updates that status as time passes. In addition you must create a main class called `YourlastnameProgram5.java` (with your actual last name) that runs the simulation given input from the user.

Your `Ball` class should have the following private fields, all of which should be `doubles`:

- `hDist` - the horizontal distance from the point where the ball was thrown in meters.
- `vDist` - the vertical distance of the ball from the ground in meters.
- `hSpeed` - how fast the ball is travelling horizontally in meters per second.
- `vSpeed` - how fast the ball is travelling vertically in meters per second.

Your `Ball` class should also have the the following public methods:

`public void initialize(double angle, double velocity, double height)` - sets up the initial conditions as follows:

- The horizontal distance should be set to 0.
- The vertical distance should be set to `height`.

- The horizontal speed should be set to `velocity` multiplied by `Math.cos(Math.toRadians(angle))`.
- The vertical speed should be set to `velocity` multiplied by `Math.sin(Math.toRadians(angle))`.

We will discuss the `Math` class in more detail when we get to Chapter 6. For now, just use the expressions above and you should be fine.

`public void update(double time)` - updates the status of the ball as follows:

- The horizontal distance traveled in a given amount of time is calculated by multiplying the time by the horizontal speed. Take this value and add it to the horizontal distance.
- We'll assume there is no wind resistance, so the horizontal speed stays the same.
- The vertical distance and speed is a bit tricker because we have to factor in the effects of gravity. We can estimate this using the following algorithm:
    1. Calculate the new vertical speed by multiplying `9.8` (the acceleration due to gravity) by the time, and subtracting it from the old vertical speed. Store this in a local variable for now, because we'll need the old value for the next step.
    2. Calculate the new vertical distance by averaging the old vertical speed and the new vertical speed, multiplying that by the time, and adding the result to the vertical distance.
    3. Set the vertical speed to the result from step 1.

`public double getHDist()` - accessor method that returns the current horizontal distance.

`public double getVDist()` - accessor method that returns the current vertical distance.

In your main Java Class, write code in the `main` method that does the following:

1. Displays a welcome message to the user.
2. Prompts for the launch angle in degrees.
3. Prompts for the initial velocity in meters/second.
4. Prompts for the initial height in meters.
5. Prompts for the time interval in seconds.
6. Creates a `Ball` object.
7. Calls the `initialize` method on the `Ball` object using the first three values the user entered.

8. Repeatedly call the `update` method on the `Ball` object using the time interval the user entered, as long as the ball is in the air (vertical distance is greater than zero.)

9. Display the horizontal distance the ball travelled to the user.

Example Input and Output

Example run 1:

```
Welcome to Nicholas Coleman's ball simulator!

Please enter the angle in degrees: 45
Please enter the initial velocity: 20
Please enter the initial height: 2
Please enter the time interval: 1

Distance traveled: 56.568542494923804 meters.
```

Example run 2:

```
Welcome to Nicholas Coleman's ball simulator!

Please enter the angle in degrees: 30
Please enter the initial velocity: 20
Please enter the initial height: 2
Please enter the time interval: 1

Distance traveled: 51.96152422706632 meters.
```

Example run 3:

```
Welcome to Nicholas Coleman's ball simulator!

Please enter the angle in degrees: 30
Please enter the initial velocity: 10
Please enter the initial height: 2
Please enter the time interval: 1

Distance traveled: 17.320508075688775 meters.
```

Notes and Comments

Upload your Java source files to the dropbox named **Program 5**. The name of the source file for the main class must be your last name followed by `Program5` with the extension `.java`. For example, mine would be `ColemanProgram5.java`.

Make sure to include comments with your name, a description of what the program does, the course (CSCI 1015), and the assignment name (Program 5) at the beginning of all of

your source files. Also make sure to add comments at the beginning of each of the methods you write describing what they do.

Make sure you only hand in the source files for your assignment, not the class files, or any other files that NetBeans creates.

Your programs must compile without errors in order to be graded. Once your program compiles make sure to test it using **multiple test cases** to see if it meets the requirements of the assignment.