# Introduction to Programming in Python

Assignment 1 (Straightline Programs) Discussion

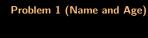
## Problem 1 (Name and Age)

☑ name.age.py		
Command-line input	name (str) and age (str)	
Standard output	a message containing name and age	

```
>_ "/workspace/straightline_programs

$ python3 name_age.py Alice 19
Alice is 19 years old.
$ python3 name_age.py Bob 23
```

Bob is 23 years old.



Accept  $name\ (str)\ and\ age\ (str)\ as\ command-line\ arguments$ 

Set message to the value "name is age years old."

Write message to standard output

#### Problem 2 (Greet Three)

#### greet\_three.py Command-line input name<sub>1</sub> (str), name<sub>2</sub> (str), and name<sub>3</sub> (str) Standard output a message containing name<sub>1</sub>, name<sub>2</sub>, and name<sub>3</sub>

```
>_ ~/workspace/straightline_programs
```

- \$ python3 greet\_three.py Alice Bob Carol Hi Carol, Bob, and Alice. \$ python3 greet\_three.py Dan Eve Fred
- Hi Fred, Eve, and Dan.



Accept  $\mathit{name}_1$  (str),  $\mathit{name}_2$  (str), and  $\mathit{name}_3$  (str) as command-line arguments

Set message to the value "Hi  $\textit{name}_3$ ,  $\textit{name}_2$ , and  $\textit{name}_1$ ."

Write message to standard output

#### Problem 3 (Day of the Week)

# Command-line input m (int), d (int), and y (int) Standard output day of the week (0 for Sunday, 1 for Monday, etc.)

#### Problem 3 (Day of the Week)

Accept m (int), d (int), and y (int) as command-line arguments

Compute dow (day of week) as follows

$$y_0 = y - (14 - m)/12$$

$$x_0 = y_0 + y_0/4 - y_0/100 + y_0/400$$

$$m_0 = m + 12 \times ((14 - m)/12) - 2$$

$$dow = (d + x_0 + 31 \times m_0/12) \mod 7$$

Write dow to standard output

#### Problem 4 (Three Sort)

# Command-line input x (int), y (int), and z (int) Standard output the numbers in sorted order

```
>_ "/workspace/straightline_programs
$ python3 three_sort.py 1 3 2
1 2 3
$ python3 three_sort.py 3 2 1
1 2 3
```

Problem 4 (Three Sort)

Accept x (int), y (int), and z (int) as command-line arguments

Set *alpha* to the smallest of the three numbers

Set omega to the largest of the three numbers

Set middle to the middle value obtained as an arithmetic combination of x, y, z, alpha, and omega

Write "alpha middle omega" to standard output

## Problem 5 (Body Mass Index)

Command-line input	w (float) and $h$ (float)
Standard output	body mass index

## >\_ ~/workspace/straightline\_programs

\$ python3 bmi.py 75 1.83 22.395413419331717 \$ python3 bmi.py 97 1.75 31.6734693877551

# Problem 5 (Body Mass Index)

Accept w (float) and h (float) as command-line arguments

Set bmi to the body mass index value computed as

Write bmi to standard output

### Problem 6 (Wind Chill)

#### 

```
>_ "/workspace/straightline_programs

$ python3 wind_chill.py 32 15
21.588988890532022

$ python3 wind_chill.py 10 10
```

## Problem 6 (Wind Chill)

Accept t (float) and v (float) as command-line arguments

Set w to the wind chill value computed as

$$w = 35.74 + 0.6215t + (0.4275t - 35.75)v^{0.16}$$

Write w to standard output

#### **Problem 7 (Gravitational Force)**

#### 

```
>_ ~/workspace/straightline_programs
```

- \$ python3 gravitational\_force.py 2e30 6e24 1.5e11
- 3.559466666666664e+22
- \$ python3 gravitational\_force.py 6e24 7.35e22 3.84e8
- 1.9960083007812498e+20

## Problem 7 (Gravitational Force)

Accept  $m_1$  (float),  $m_2$  (float), and r (float) as command-line arguments

Set G to  $6.674 \times 10^{-11}$ 

Set f to the gravitational force value computed as

$$f=G\frac{m_1m_2}{r^2}$$

Write f to standard output

#### Problem 8 (Gambler's Ruin)

#### @ gambler.py $n_1$ (int), $n_2$ (int), and p (float) Command-line input Standard output probabilities $p_1$ and $p_2$

## >\_ ~/workspace/straightline\_programs \$ python3 gambler.py 10 100 0.51

- 0.6661883734200654 0.3338116265799349
- \$ python3 gambler.py 100 10 0.51
- 0.006110712510580903 0.9938892874894192

## Problem 8 (Gambler's Ruin)

Accept  $n_1$  (int),  $n_2$  (int), and p (float) as command-line arguments

Set q to 1-p

Set  $p_1$  and  $p_2$  to probability values computed as

$$p_1=rac{1-(rac{
ho}{q})^{n_2}}{1-(rac{
ho}{q})^{n_1+n_2}} ext{ and } p_2=rac{1-(rac{q}{p})^{n_1}}{1-(rac{q}{p})^{n_1+n_2}}$$

Write " $p_1 p_2$ " to standard output

### Problem 9 (Waiting Time)

#### @ waiting\_time.py $\lambda$ (float) and t (float) Command-line input Standard output probability p

```
>_ ~/workspace/straightline_programs
$ python3 waiting_time.py 0.1 5
```

- \$ python3 waiting\_time.py 0.6 3

## Problem 9 (Waiting Time)

Accept  $\lambda$  (float) and t (float) as command-line arguments

Set p to the probability value computed as

$$=e^{-\lambda t}$$

Write p to standard output

#### Problem 10 (Cartesian Coordinates)

☑ cartesian.py	
Command-line input	r (float) and $ heta$ (float)
Standard output	cartesian values $x$ and $y$

#### >\_ ~/workspace/straightline\_programs

- \$ python3 cartesian.py 1 45
  0.7071067811865476 0.7071067811865475
  \$ python3 cartesian.py 1 60

## Problem 10 (Cartesian Coordinates)

Accept r (float) and  $\theta$  (float) as command-line arguments

Set x and y to the cartesian values computed as

$$x = r \cos(\theta)$$
 and  $y = r \sin(\theta)$ 

Write "x y" to standard output

#### **Problem 11 (Great Circle Distance)**

#### 

```
>_ "/workspace/straightline.programs

$ python3 great_circle.py 48.87 -2.33 37.8 -122.4
8701.387455462233
$ python3 great_circle.py 46.36 -71.06 39.90 116.41
10376.503884802196
```

### **Problem 11 (Great Circle Distance)**

Accept  $x_1$  (float),  $y_1$  (float),  $x_2$  (float), and  $y_2$  (float) as command-line arguments

Set d to the great circle distance value computed as

$$d = 6359.83 \arccos(\sin(x_1)\sin(x_2) + \cos(x_1)\cos(x_2)\cos(y_1 - y_2))$$

Write d to standard output

## Problem 12 (Snell's Law)

🕝 snell.py	
Command-line input	$ heta_1$ (float), $n_1$ (float), and $n_2$ (float)
Standard output	angle of refraction

```
>_ "/workspace/straightline_programs

$ python3 snell.py 58 1 1.52

33.912513998258994

$ python3 snell.py 30 1 1.2

24.624318352164074
```

## Problem 12 (Snell's Law)

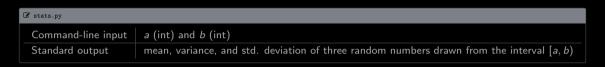
Accept  $\theta_1$  (float),  $n_1$  (float), and  $n_2$  (float) as command-line arguments

Set  $\theta_2$  to the angle of refraction value computed as

$$heta_2 = \arcsin\left(rac{n_1}{n_2}\sin( heta_1)
ight)$$

Write  $\theta_2$  to standard output

#### **Problem 13 (Uniform Random Numbers)**



#### >\_ ~/workspace/straightline\_programs

- \$ python3 stats.py 0 1
  0.5731084550427492 0.04897843881307027 0.22131072909615176
- \$ python3 stats.py 50 100
- 91.3736830296877 25.288830238538182 5.028800079396494

#### **Problem 13 (Uniform Random Numbers)**

Accept a (int) and b (int) as command-line arguments

Set  $x_1$ ,  $x_2$ , and  $x_3$  to random numbers drawn from the interval [a, b]

Set  $\mu$ , var, and  $\sigma$  to the mean, variance, and std. deviation values computed as

$$\mu = (x_1 + x_2 + x_3)/3$$
,  $var = ((x_1 - \mu)^2 + (x_2 - \mu)^2 + (x_3 - \mu)^2)/3$ , and  $\sigma = \sqrt{var}$ 

Write " $\mu$  var  $\sigma$ " to standard output

#### Problem 14 (Die Roll)

# Command-line input | n (int) Standard output | sum of two n-sided die rolls

```
>_ "/workspace/straightline_programs

$ python3 die_roll.py 6
12
$ python3 die_roll.py 6
10
```

## Problem 14 (Die Roll)

Accept n (int) as command-line argument

Set  $\mathit{die}_1$  and  $\mathit{die}_2$  to random integers drawn from the interval [1,n]

Write  $die_1 + die_2$  to standard output

#### **Problem 15 (Triangle Inequality)**

# Command-line input |x| (int), |y| (int), and |z| (int) Standard output |x| true if each is less than or equal to the sum of the other two, and False otherwise

```
>_ "/workspace/straightline_programs

$ python3 triangle.py 3 3 3
True

$ python3 triangle.py 2 4 7
```

F	Problem 15 (Triangle Inequality)		
	Accept $x$ (int), $y$ (int), and $z$ (int) as command-line arguments		
	Set $expr$ to a boolean expression which is $True$ if each of $x$ , $y$ , and $z$ is less than or equal to the sum of the other two, and $False$ otherwise		

Write expr to standard output