CSIT110 / CSIT810
Python

Lecture 8

Dr. Joseph Tonien

School of Computing and Information Technology
University of Wollongong

Objectives

More on:

- String data type
- Boolean data type

Problem solving using:

- loop
- function

String data type

String data type

Find the length of a string:

```
greeting = "Hi there!"
greeting_length = len(greeting)  # → 9
```

Get substrings, one character at a time:

```
print(greeting[0])  # → H
print(greeting[1])  # → i
print(greeting[2])  # → space
print(greeting[3])  # → t
print(greeting[4])  # → h
print(greeting[5])  # → e
print(greeting[6])  # → r
print(greeting[7])  # → e
print(greeting[8])  # → !
```

String data type

```
greeting = "Hi there!"

for i in range(0, len(greeting)):
  print(greeting[i])
```

What is this code doing?

String data type

Slicing a string:

```
s = "Python is cool!"

s1 = s[1:4]
# now s1 = "yth"
```

s[i:j] gives substring from index i up to index (j-1), so altogether, there are (j-i) characters

```
s2 = s[1:]
# now s2 = "ython is cool!"
```

s[i:] gives substring from index i up to the end

```
s3 = s[:4]
# now s1 = "Pyth"
```

s[:j] gives substring from index 0 up to index (j-1), so altogether, there are j characters

String data type

Upper case:

```
name = "John Smith"
name_uppercase = name.upper()
print(name_uppercase)
```

```
JOHN SMITH
```

Lower case:

```
name = "John Smith"
name_lowercase = name.lower()
print(name_lowercase)
```

```
john smith
```

Boolean data type

Boolean data type

We have learned about:

- Number data type
- String data type

Now, we will learn a new data type called **Boolean**

Example:

```
has_discount_coupon = True

brownies_serve_hot = False

virus_found = True

file_scan_completed = False
```

Boolean data type

Example:

```
# ask the user how brownies to be served
brownies_input = input("Would you like your chocolate
brownies served hot or cold: ")

if (brownies_input == "hot"):
  brownies_serve_hot = True
else:
  brownies_serve_hot = False
```

Belgian choc fudge brownies

Served hot or cold

Boolean data type

Example:

```
# ask the user if they want to watch additional star show
star_show_input = input("Add Stargazing Show: (Y/N) ")
if ((star_show_input == "Y") or (star_show_input == "y")):
  include_star_show = True
else:
  include_star_show = False
```

Rewrite Science Park Program using functions

```
# print menu function
def print_menu():
  print("------------------------------------------------")
  print("          Welcome to Science Park!              ")
  print()
  print("Admission Charges: Adult $35, Child $20         ")
  print("Stargazing Show: $10/person                     ")
  print()
  print("Free Science Park Hats if you spend $150 or more")
  print("10% discount if you spend $200 or more          ")
  print("------------------------------------------------")
  print()
```

```
# take order from user, this function returns three values:
# (1) number of adults (2) number of children
# (3) include star show or not
def take_order():
  print("Please make your order.")
  print()

  # ask number of adults
  adult_input = input("Enter number of adults: ")
  adult = int(adult_input)

  # ask number of children
  child_input = input("Enter number of children: ")
  child = int(child_input)

  # ask the additional star show
  star_show_input = input("Add Stargazing Show: (Y/N) ")
  if ((star_show_input == "Y") or (star_show_input == "y")):
    include_star_show = True
  else:
    include_star_show = False

  return adult, child, include_star_show
```

```
ADULT_PRICE = 35
CHILD_PRICE = 20
SHOW_PRICE = 10
DISCOUNT_MIN = 200 # the minimum amount to have discount
DISCOUNT_PCT = 10  # the discount percentage
FREE_HAT_MIN = 150 # the minimum amount to have free hat

# print cost
def print_cost(adult, child, include_star_show):
  # calculate the total charge, no discount calculation yet
  adult_cost = ADULT_PRICE * adult
  child_cost = CHILD_PRICE * child

  if (include_star_show):
    show_cost = SHOW_PRICE * (adult + child)
  else:
    show_cost = 0

  total_cost = adult_cost + child_cost + show_cost
```

(continued next page)

```
# calculate discount and final charge
if (total_cost >= DISCOUNT_MIN):
  # eligible for discount
  final_cost = total_cost * (100 - DISCOUNT_PCT) / 100

  print("Total cost: ${0}".format(total_cost))
  print("Discount {0}%".format(DISCOUNT_PCT))
  print("Final charge: ${0}".format(final_cost))
else:
  # not eligible for discount
  final_cost = total_cost

  print("Final charge: ${0}".format(final_cost))

# check Free Hat
if (total_cost >= FREE_HAT_MIN):
  print("Please collect Science Park Hats at the counter.")

print()
print("Enjoy your day!!!")
```

```
# main program

# print menu to user
print_menu()

# take order from user
adult, child, include_star_show = take_order()

# print receipt to user
print_cost(adult, child, include_star_show)
```

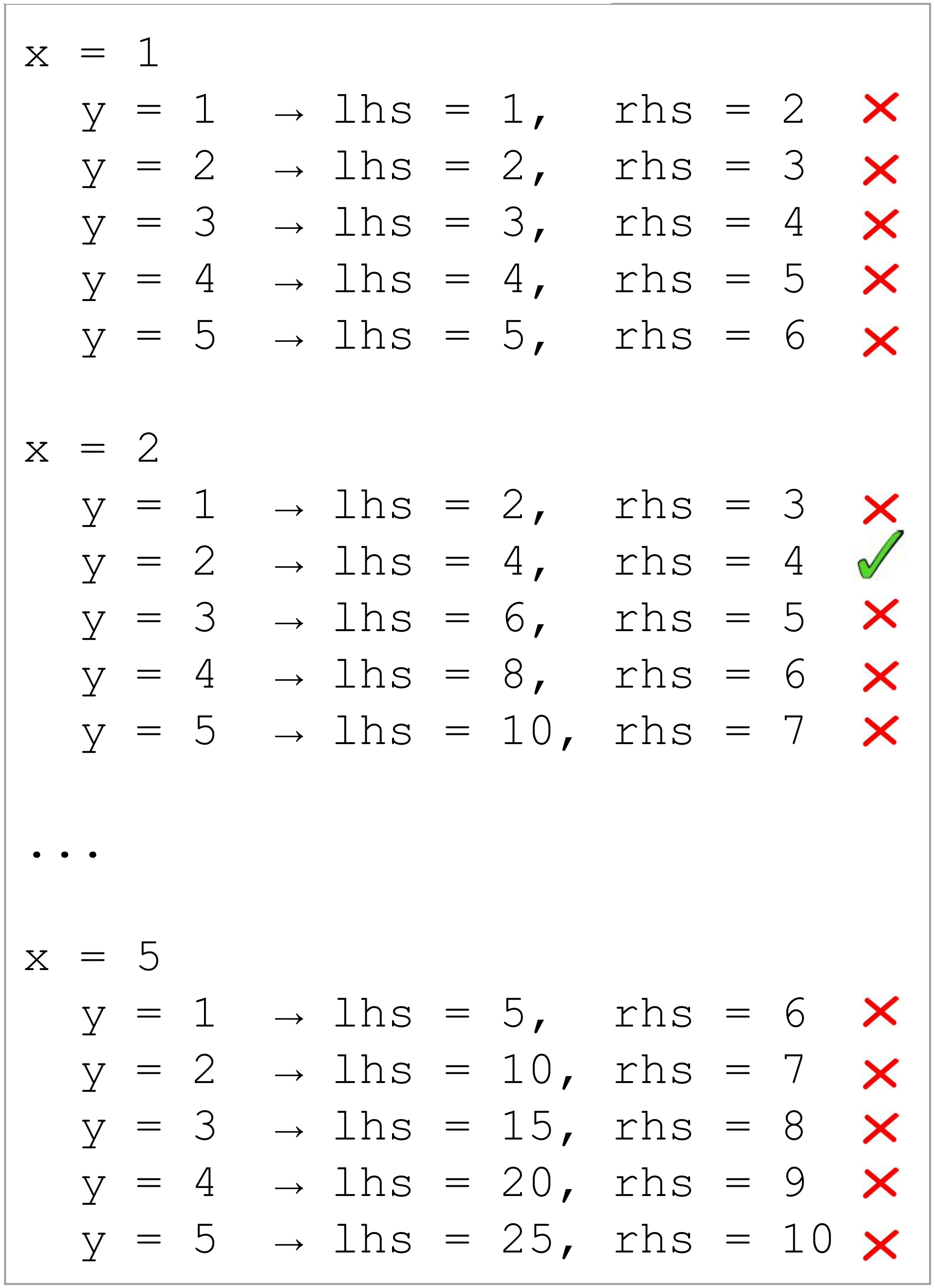
Find all positive integers x and y such that

$$1 \leq x \leq 5$$

$$1 \leq y \leq 5$$

$$x \times y = x + y$$

$x \times y = x + y$???



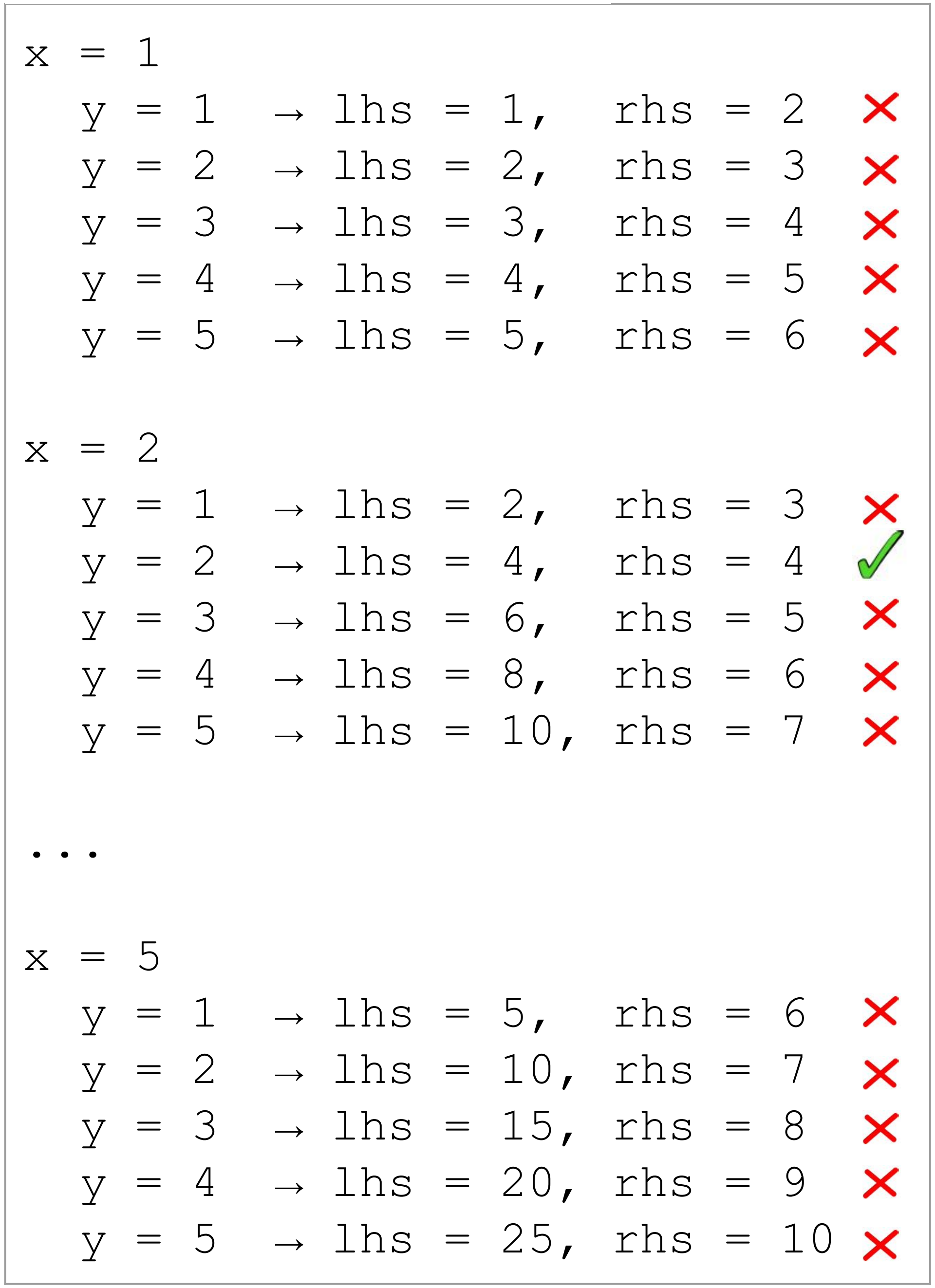
```
x = 1
   y = 1  → lhs = 1,  rhs = 2  ✗
   y = 2  → lhs = 2,  rhs = 3  ✗
   y = 3  → lhs = 3,  rhs = 4  ✗
   y = 4  → lhs = 4,  rhs = 5  ✗
   y = 5  → lhs = 5,  rhs = 6  ✗

x = 2
   y = 1  → lhs = 2,  rhs = 3  ✗
   y = 2  → lhs = 4,  rhs = 4  ✓
   y = 3  → lhs = 6,  rhs = 5  ✗
   y = 4  → lhs = 8,  rhs = 6  ✗
   y = 5  → lhs = 10, rhs = 7  ✗

...

x = 5
   y = 1  → lhs = 5,  rhs = 6  ✗
   y = 2  → lhs = 10, rhs = 7  ✗
   y = 3  → lhs = 15, rhs = 8  ✗
   y = 4  → lhs = 20, rhs = 9  ✗
   y = 5  → lhs = 25, rhs = 10 ✗
```

$x \times y = x + y$???

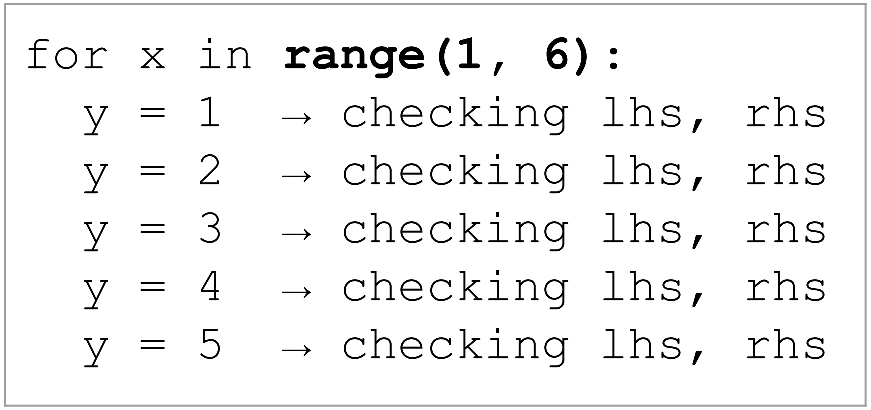


```
x = 1
   y = 1  → lhs = 1,  rhs = 2  ✗
   y = 2  → lhs = 2,  rhs = 3  ✗
   y = 3  → lhs = 3,  rhs = 4  ✗
   y = 4  → lhs = 4,  rhs = 5  ✗
   y = 5  → lhs = 5,  rhs = 6  ✗

x = 2
   y = 1  → lhs = 2,  rhs = 3  ✗
   y = 2  → lhs = 4,  rhs = 4  ✓
   y = 3  → lhs = 6,  rhs = 5  ✗
   y = 4  → lhs = 8,  rhs = 6  ✗
   y = 5  → lhs = 10, rhs = 7  ✗


...


x = 5
   y = 1  → lhs = 5,  rhs = 6  ✗
   y = 2  → lhs = 10, rhs = 7  ✗
   y = 3  → lhs = 15, rhs = 8  ✗
   y = 4  → lhs = 20, rhs = 9  ✗
   y = 5  → lhs = 25, rhs = 10 ✗
```



```
for x in range(1, 6):
   y = 1  → checking lhs, rhs
   y = 2  → checking lhs, rhs
   y = 3  → checking lhs, rhs
   y = 4  → checking lhs, rhs
   y = 5  → checking lhs, rhs
```

$x \times y = x + y$???

```
x = 1
  y = 1  → lhs = 1,  rhs = 2  ✗
  y = 2  → lhs = 2,  rhs = 3  ✗
  y = 3  → lhs = 3,  rhs = 4  ✗
  y = 4  → lhs = 4,  rhs = 5  ✗
  y = 5  → lhs = 5,  rhs = 6  ✗

x = 2
  y = 1  → lhs = 2,  rhs = 3  ✗
  y = 2  → lhs = 4,  rhs = 4  ✓
  y = 3  → lhs = 6,  rhs = 5  ✗
  y = 4  → lhs = 8,  rhs = 6  ✗
  y = 5  → lhs = 10, rhs = 7  ✗


...


x = 5
  y = 1  → lhs = 5,  rhs = 6  ✗
  y = 2  → lhs = 10, rhs = 7  ✗
  y = 3  → lhs = 15, rhs = 8  ✗
  y = 4  → lhs = 20, rhs = 9  ✗
  y = 5  → lhs = 25, rhs = 10 ✗
```

```python
for x in range(1, 6):
  for y in range(1, 6):
    # checking lhs, rhs
```

$x \times y = x + y$???

```
for x in range(1, 6):
  for y in range(1, 6):
    # checking lhs, rhs
    lhs = x * y
    rhs = x + y
    if (lhs == rhs):
      print("Solution found x = {0}, y = {1}".format(x, y))
```

```
Solution found x = 2, y = 2
```

Find all positive integers x and y such that

$$1 \leq x \leq 100$$

$$1 \leq y \leq 100$$

$$(x + y)^5 = x^5 + y^5 + 3\,000\,000$$

$$1 \leq x \leq 100$$

$$1 \leq y \leq 100$$

$$(x+y)^5 = x^5 + y^5 + 3\,000\,000$$

```
for x in range(1, 101):
  for y in range(1, 101):
    # checking lhs, rhs
    lhs = (x + y)**5
    rhs = x**5 + y**5 + 3000000
    if (lhs == rhs):
      print("Solution found x = {0}, y = {1}".format(x, y))
```

```
Solution found x = 10, y = 10
```

Find all positive integers x, y and z such that

$$1 \leq x \leq 100$$

$$1 \leq y \leq 100$$

$$1 \leq z \leq 100$$

$$x + y + z = x \times y \times z$$

$$1 \leq x \leq 100$$

$$1 \leq y \leq 100$$

$$1 \leq z \leq 100$$

$$x + y + z = x \times y \times z$$

```
for x in range(1, 101):
  for y in range(1, 101):
    for z in range(1, 101):
      # checking lhs, rhs
      lhs = x + y + z
      rhs = x * y * z
      if (lhs == rhs):
        print("Solution found x = {0}, y = {1}, z = {2}".format(x, y, z) )
```

```
Solution found x = 1, y = 2, z = 3
Solution found x = 1, y = 3, z = 2
Solution found x = 2, y = 1, z = 3
Solution found x = 2, y = 3, z = 1
Solution found x = 3, y = 1, z = 2
Solution found x = 3, y = 2, z = 1
```

Guessing Game

```
I have chosen a secret number between 0 and 100.

Make a guess: 50
Lower than that

Make a guess: 25
Higher than that

Make a guess: 30
Lower than that

Make a guess: 27
Lower than that

Make a guess: 26
Hooray! You are correct!
```

Guessing Game

```
print("I have chosen a secret number between 0 and 100.")

while True:
  print()
  guess_input = input("Make a guess: ")
  guess = int(guess_input)

  if (guess == secret):
    print("Hooray! You are correct!")
    break
  elif (guess > secret):
    print("Lower than that")
  else:
    print("Higher than that")
```

The 3x + 1 problem

https://www.youtube.com/watch?v=5mFpVDpKX70

$$\begin{tikzcd}
 & x/2 & \text{if } x \text{ is even} \\
x \arrow[ur] \arrow[dr] & & \\
 & 3x + 1 & \text{if } x \text{ is odd}
\end{tikzcd}$$

$$\begin{tikzcd}
10 \arrow[r] & 5 \arrow[r] & 16 \arrow[d] & & & \\
 & & 8 \arrow[r] & 4 \arrow[r] & 2 \arrow[r] & 1
\end{tikzcd}$$

Collatz Conjecture: Every positive integer will eventually lead to 1

10 → 5 → 16 → 8 → 4 → 2 → 1

We want to write a program to test the conjecture

```
Enter a positive integer: 10
Step 0: 10
Step 1: 5
Step 2: 16
Step 3: 8
Step 4: 4
Step 5: 2
Step 6: 1
```

10 → 5 → 16 → 8 → 4 → 2 → 1

```
Enter a positive integer: 10
Step 0: 10
Step 1: 5
Step 2: 16
Step 3: 8
Step 4: 4
Step 5: 2
Step 6: 1
```

```
number_input = input("Enter a positive integer: ")
n = int(number_input)

print_steps(n)
```

We need to write a function to print out the step details

10 → 5 → 16 → 8 → 4 → 2 → 1

```
Enter a positive integer: 10
Step 0: 10
Step 1: 5
Step 2: 16
Step 3: 8
Step 4: 4
Step 5: 2
Step 6: 1
```

```
def print_steps(number):
  step = 0
  while True:
    print("Step {0}: {1}".format(step, number))
    ...
```

10 → 5 → 16 → 8 → 4 → 2 → 1

```
Enter a positive integer: 10
Step 0: 10
Step 1: 5
Step 2: 16
Step 3: 8
Step 4: 4
Step 5: 2
Step 6: 1
```

```
def print_steps(number):
  step = 0
  while True:
    print("Step {0}: {1}".format(step, number))

    # if we reach 1 then stop the loop
    if (number == 1):
      break


    ...
```

10 → 5 → 16 → 8 → 4 → 2 → 1

```
Enter a positive integer: 10
Step 0: 10
Step 1: 5
Step 2: 16
Step 3: 8
Step 4: 4
Step 5: 2
Step 6: 1
```

```
def print_steps(number):
  step = 0
  while True:
    print("Step {0}: {1}".format(step, number))

    # if we reach 1 then stop the loop
    if (number == 1):
      break

    # next step
    if (number%2 == 0):
      number = number // 2
    else:
      number = 3 * number + 1

    step += 1
```

```
def print_steps(number):
  step = 0
  while True:
    print("Step {0}: {1}".format(step, number))

    # if we reach 1 then stop the loop
    if (number == 1):
      break

    # next step
    if (number%2 == 0):
      number = number // 2
    else:
      number = 3 * number + 1

    step += 1

# main program
number_input = input("Enter a positive integer: ")
n = int(number_input)

print_steps(n)
```