Lesson 1   Arithmetic with Python

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

New Commands

```python
print("what's your name?")
print(5 + 6)
```
Procedures

Task 1: Translate the mathematical expressions below into Python code. Use Python to calculate the results.

<table>
<thead>
<tr>
<th>Example</th>
<th>Symbol</th>
<th>Name</th>
<th>Python Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 + 6 = 11</td>
<td>+</td>
<td>Addition</td>
<td>+</td>
</tr>
<tr>
<td>11 – 5 = 6</td>
<td>-</td>
<td>Subtraction</td>
<td>-</td>
</tr>
<tr>
<td>5 × 2 = 10</td>
<td>×</td>
<td>Multiplication</td>
<td>*</td>
</tr>
<tr>
<td>10 ÷ 5 = 2</td>
<td>÷</td>
<td>Division</td>
<td>/</td>
</tr>
<tr>
<td>5² = 25</td>
<td>xⁿ</td>
<td>Exponentiation</td>
<td>**</td>
</tr>
<tr>
<td>11 mod 5 = 1</td>
<td></td>
<td>Modulo</td>
<td>%</td>
</tr>
</tbody>
</table>

Task 2: Calculate the results of the mathematical expressions below:

181 + 125 + 669 = ?
2160 - 439 - 57 = ?
79 × 21 × 3 × 108 = ?
40257 ÷ 1917 = ?
156⁴ = ?
40257 modulo 41 = ?

Task 3: Suppose your current computing skill level is 100. If you spend 1 hour per day in practising Python, your skill level will increase by 1% per day.
Create Python code to calculate the level of your computing skills if you keep doing this for 7 days. What will happen then after 180 days (and 365 days)?
Lesson 2  Text and Numbers in Python

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

New Commands

```python
print(type("What's your name?"))

print(type(5 + 6))

print(type(40257 / 1917))

input("What's your name? Your answer: ")
```
Procedures

Task 1: Calculate the expressions below and determine the type of data of the results:

100000 + 365.12 = ?
1989.12 - 917.8 = ?
36.6 × 50.1 = ?
100.25 × 40 = ?
3.310 = ?
2501 mod 2.5 = ?

Task 2: Read the script below and identify a new function:

1. print(type(input("What's your name? Your answer: ")))
2. print(type(input("How old are you? Your answer: ")))
3. print(type(input("How tall are you? Your answer:(metre) ")))

Task 3: Modify the script below and convert the data types of the input information:

1. print(type(input("Your school's postcode: ")))
2. print(type(input("Your school's name: ")))
3. print(type(input("Street/Block number nearby: ")))
4. print(type(input("Your body temperature:(°C) ")))
5. print(type(input("Today's temperature:(°C) ")))
6. print(type(input("Your lucky number: ")))}
Lesson 3  Triangle Area Calculator

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

New Commands

```python
my_var = 1
my_var = my_var + 1
```
Procedures

Step 1  Use

Read and run the program.

```python
1. print("Triangle Area Calculator")
2.
3. a = float(input("Value of side a:(centimetre) "))
4. b = float(input("Value of base b:(centimetre) "))
5. c = float(input("Value of side c:(centimetre) "))
6.
7. s = (a + b + c) / 2
8.
9. area = (s * (s-a) * (s-b) * (s-c)) ** 0.5
10.
11. print("The area of this triangle is", area, "square centimetres.")
```

Questions:

- Why is `float()` used together with `input()`?
- How can the program remember the values of three sides of a triangle?
- How to calculate the semi-perimeter by using Python?
- How to calculate the area by using Python?
- Look at Line 11. Do you notice anything different?
Step 2  Modify

Task 1: Create another Triangle Area Calculator that calculates the area of a triangle with the given base and height.

Task 2: Based on the program for Triangle Area Calculator, create a Triangular Prism Volume Calculator.

Task 3: Create an application to calculate the capacity of a milk carton. Remember, you need to use the unit of measurement for liquids.

Step 3  Create

Create a calculator application that can calculate the perimeter and area of a plane shape or the volume of a solid shape, for instance a prism or prism-like figure.
Lesson 4  Body Mass Index Calculator

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

New Commands

```python
if BMI >= 30.0:
    # Code

if BMI >= 18.5 and BMI <= 24.9:
    # Code
```
**Procedures**

### BMI Healthy Weight (Adults)

<table>
<thead>
<tr>
<th>BMI</th>
<th>Level</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Below 18.5</td>
<td>Underweight</td>
<td>Being underweight may lead to a weakened immune system and feeling tired.</td>
</tr>
<tr>
<td>18.5 – 24.9</td>
<td>Healthy Weight</td>
<td>You have a healthy weight for your height.</td>
</tr>
<tr>
<td>25.0 – 29.9</td>
<td>Overweight</td>
<td>You are heavier than is healthy for someone of your height. Try to lose weight by keeping a balanced diet and physical activity.</td>
</tr>
<tr>
<td>30.0 and Above</td>
<td>Obese</td>
<td></td>
</tr>
</tbody>
</table>

*Source: UK NHS, US CDC*

### Calculate the BMI

<table>
<thead>
<tr>
<th>Name</th>
<th>Height (m)</th>
<th>Weight (kg)</th>
<th>BMI</th>
<th>Weight Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Your Name)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ai Fukuhara</td>
<td>1.55</td>
<td>48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Angelina Jolie</td>
<td>1.69</td>
<td>54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arnold Schwarzenegger</td>
<td>1.88</td>
<td>113</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Benedict Cumberbatch</td>
<td>1.83</td>
<td>79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jackie Chan</td>
<td>1.70</td>
<td>65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kobe Bryant</td>
<td>1.98</td>
<td>96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malgorzata Dydek</td>
<td>2.18</td>
<td>101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maya Plisetskaya</td>
<td>1.77</td>
<td>59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sarah Hyland</td>
<td>1.57</td>
<td>45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Serena Williams</td>
<td>1.75</td>
<td>70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tom Cruise</td>
<td>1.70</td>
<td>77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yao Ming</td>
<td>2.29</td>
<td>141</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zoe</td>
<td>1.70</td>
<td>72</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step 1  Predict

Read the program.

```python
1. print("BMI Calculator")
2.
3. height = float(input("Height:(metre) "))
4. weight = float(input("Weight:(kilogram) "))
5.
6. BMI = weight / (height**2)
7. print(BMI)
8.
9. if BMI >= 30.0:
10.   print("Obese")
11.
12. if BMI >= 25.0 and BMI <= 29.9:
13.   print("Overweight")
14.
15. if BMI >= 18.5 and BMI <= 24.9:
16.   print("Healthy Weight")
17.
18. if BMI < 18.5:
19.   print("Underweight")
```

Step 2  Run

Run the program.
Step 3    Investigate

Task 1: Comment on the code. Briefly explain the function.

Task 2: Investigate the conditions of the statements. Why does the program use the ‘≥’ and ‘≤’ operators to define the lowest and/or highest values of a range?

Task 3: Investigate the indentation. Remove the leading whitespace before the

```
print("obese")/print("overweight")/print("healthy weight")/print("underweight"). Run the program and see what happens. Does the program still work?
```

Task 4: Calculate Zoe’s BMI and output her weight status. Could the program decide Zoe’s weight status? Why or why not?

Step 4    Modify and Make

Create a BMI Calculator for children and teenagers.

References:

The BMI Calculator created by the UK’s NHS:  https://www.nhs.uk/live-well/healthy-weight/bmi-calculator

The BMI Calculator created by the US’s CDC:

For children and teenagers:  https://www.cdc.gov/healthyweight/bmi/calculator.html

For adults:

Lesson 5     Buy Low, Sell High

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
</table>

New Commands

```python
import random

def random.randint(start, end, step):
    # ...:

elif <...>:

else:
```
Procedures

Step 1   Predict

Read the program.

```python
1. import random
2.
3. price_a = random.randint(100, 200)
4. print("The traded price now is", price_a)
5.
6. order = int(input("Your posted price: "))
7.
8. price_b = random.randint(100, 200)
9. print("The traded price now is", price_b)
10.
11. if order > price_b:
12.     print("Current price is lower than your posted price. Buy it.")
13.
14. elif order < price_b:
15.     print("Current price is higher than your posted price. Sell it.")
16.
17. else:
18.     print("No transaction.")
```

Questions:

- What is the control structure?
- What does ‘import random’ mean?
- What are the values of the variable ‘price_a’ and ‘price_b’?

Step 2   Run

Run the program.
Step 3   Investigate

Task 1: Comment on the code. Briefly explain the function.

Task 2: Investigate the conditions of the statements. How does the program compare the posted price with the traded price?

Task 3: What is the condition of ‘else’. Write down its condition.

Step 4   Modify

Task 4: Use the if statements instead to rewrite the example program.

Step 5   Make

Create a digital artefact with conditionals that helps with decision making.
Lesson 6  Repay Loans

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
</table>

New Commands

```python
for i in range(start, end, step):
    round()
```
Procedures

Step 1  Predict

Read the program.

<table>
<thead>
<tr>
<th>Step</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>print(&quot;Loan Payment Calculator: Even Principal Payments&quot;)</td>
</tr>
<tr>
<td>2.</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>loan = int(input(&quot;Loan Amount (GBP £): &quot;))</td>
</tr>
<tr>
<td>4.</td>
<td>term = int(input(&quot;Number of Payments (months): &quot;))</td>
</tr>
<tr>
<td>5.</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>print(&quot;The annual interest rate is 5.4%&quot;)</td>
</tr>
<tr>
<td>7.</td>
<td>rate = 0.054</td>
</tr>
<tr>
<td>8.</td>
<td>balance = loan</td>
</tr>
<tr>
<td>9.</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>for i in range(1, term + 1):</td>
</tr>
<tr>
<td>11.</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>principal_payment = round(loan / term, 2)</td>
</tr>
<tr>
<td>13.</td>
<td>interest_payment = round(balance * (rate/12), 2)</td>
</tr>
<tr>
<td>14.</td>
<td>print(&quot;Repayment&quot;, i, &quot;:&quot;, principal_payment + interest_payment)</td>
</tr>
<tr>
<td>15.</td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td>balance = loan - principal_payment*i</td>
</tr>
<tr>
<td>17.</td>
<td>print(&quot;Unpaid Balance: &quot;, balance)</td>
</tr>
</tbody>
</table>

Questions:

- What is the control structure?
- What does ‘range(1, term+1)’ mean?
- How to calculate the unpaid balance of the loan after each repayment?

Step 2  Run

Run the program.
Step 3   Investigate

Task 1: Summarise the mathematical expressions.

Task 2: Based on the result returned by the Loan Payment Calculator, create a graph to present the loan repayments.

Step 4   Modify

Task 3: Modify the program to pay the loan repayments annually.

Task 4: The interest rate can be different across different repayment terms. Add the conditionals into the program you have created in the third task.

<table>
<thead>
<tr>
<th>Term (Years)</th>
<th>Interest Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5.4%</td>
</tr>
<tr>
<td>5</td>
<td>5.6%</td>
</tr>
<tr>
<td>8</td>
<td>6.1%</td>
</tr>
<tr>
<td>10</td>
<td>5.8%</td>
</tr>
</tbody>
</table>

Step 5   Make

Create a Loan Repayment Calculator application for even total payments.
Lesson 7  Toss a Coin

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Procedures

<table>
<thead>
<tr>
<th>Total Flips</th>
<th>Heads</th>
<th>% Heads</th>
<th>Tails</th>
<th>% Tails</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Step 1  Predict

Read the program.

```python
1. import random
2.
3. countheads = 0
4. counttails = 0
5.
6. n = int(input("The number of times to throw up a coin: "))
7.
8. for i in range(0, n):
9.     result = random.randint(0, 1)
10.
11.    if result == 0:
12.        countheads += 1
13.        print(result)
14.    else:
15.        counttails += 1
16.        print(result)
17.
18. print("Result:")
19. print("Heads up: ", countheads)
20. print("Tails up: ", counttails)
21.
22. print("Chance:")
23. print("Heads : Tails", round(countheads/n, 2), ":", round(counttails/n, 2))
```

Questions:

- How does the program represent the results of heads and tails?
- How does the program count the number of heads or tails?
- Explain the use of the if...else statement and the for loop.

Step 2  Run

Run the program.
Step 3  Investigate

Answer the questions.

Step 4  Modify

Task 1: Add a line of code to calculate the probabilities of getting heads and tails.

Task 2: Change the number of the toss into a random number generated between 4040 and 80640.

Task 3: Bet on the toss of the coin. Modify the example program to ask the user to input their guess before the coin-tossing experiment. Report the probability result and the betting result.

Step 5  Make

Based on the example program, modify the functions. Suppose the program is to model the dice-rolling process, how should you edit the code?

Record the probability to get each number of the dice.
Lesson 8     Guess the Number
Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
</table>

New Commands

```python
while True:
    break
```
Procedures

Step 1  Predict

Read the program.

```python
1. import random
2.
3. num = random.randint(1, 100)
4.
5. t = 7
6.
7. print("Guess a number between 1 and 100. You have 7 attempts.")
8. print("Let's start.")
9.
10. while True:
11.     guess = int(input("Guess the number: "))
12.     if guess == num:
13.         print("BINGO!")
14.         break
15.
16.     elif guess > num:
17.         t -= 1
18.         print("High! You have", t, "attempts left.")
19.
20.     elif guess < num:
21.         t -= 1
22.         print("Low! You have", t, "attempts left.")
23.
24.     if t == 0:
25.         print("The correct number is", num)
26.         print("Game Over!")
27.         break
```

Step 2  Run

Run the program.
### Step 3  Investigate and Modify

<table>
<thead>
<tr>
<th>Round</th>
<th>1st Try</th>
<th>2nd Try</th>
<th>3rd Try</th>
<th>4th Try</th>
<th>5th Try</th>
<th>6th Try</th>
<th>7th Try</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>7</td>
<td>85</td>
<td>60</td>
<td>40</td>
<td>20</td>
<td>11</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

### Note

The 33 reserved words in Python:

<table>
<thead>
<tr>
<th>and</th>
<th>as</th>
<th>assert</th>
<th>break</th>
<th>class</th>
<th>continue</th>
</tr>
</thead>
<tbody>
<tr>
<td>def</td>
<td>del</td>
<td>elif</td>
<td>else</td>
<td>except</td>
<td>finally</td>
</tr>
<tr>
<td>for</td>
<td>from</td>
<td>False</td>
<td>global</td>
<td>if</td>
<td>import</td>
</tr>
<tr>
<td>in</td>
<td>is</td>
<td>lambda</td>
<td>nonlocal</td>
<td>not</td>
<td>None</td>
</tr>
<tr>
<td>or</td>
<td>pass</td>
<td>raise</td>
<td>return</td>
<td>try</td>
<td>True</td>
</tr>
<tr>
<td>while</td>
<td>with</td>
<td>yield</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lesson 9~10  Python Quizzes

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
New Commands

```python
import cyberpi

cyberpi.display
cyberpi.display.clear()
cyberpi.console.print()
cyberpi.console.println()

cyberpi.led

cyberpi.led.off()
cyberpi.led.off(id="3")
cyberpi.led.on()
cyberpi.led.on("green", id="all")
cyberpi.led.on("red", id=3)
cyberpi.led.show("orange yellow cyan blue purple")
cyberpi.led.play(name="firefly")

cyberpi.controller.is_press()
cyberpi.controller.is_press("a")
cyberpi.controller.is_press("b")
```
4) **Joystick:**

```python
cyberpi.controller.is_press()
cyberpi.controller.is_press("up")
cyberpi.controller.is_press("down")
cyberpi.controller.is_press("right")
cyberpi.controller.is_press("left")
cyberpi.controller.is_press("middle")
```
Procedures

Features of CyberPi

1. Light sensor
2. WiFi + Bluetooth ESP32
3. Button A (Return key)
4. Button B (Confirmation key)
5. Gyroscope
   Accelerometer
6. Speaker
Step 1  Predict

Read the program.

```python
1. import cyberpi
2. cyberpi.display.clear()
3. cyberpi.led.off()
4. cyberpi.console.println("A - True")
5. cyberpi.console.println("B - False")
6. cyberpi.led.on(255, 255, 255)
7. cyberpi.console.println("Python is a compiled language.")
8. cyberpi.console.println("Your Answer:")
9. while True:
10.     if cyberpi.controller.is_press("a"):
11.         cyberpi.led.on(255, 0, 0)
12.         cyberpi.console.println("Incorrect.")
13.         cyberpi.console.println("Correct Answer: False")
14.         break
15.     if cyberpi.controller.is_press("b"):
16.         cyberpi.led.on(0, 255, 0)
17.         cyberpi.console.println("Correct!")
18.         break
```

Questions:

- Which module is imported? Explain why we should import this module.
- How to print text on the screen?
- How to light up/off the LED strip?
- How to set the color of the LED? How to represent the LED colors in Python?
- How to enter the answer?
- How to evaluate the answer?
Step 2  Run

Run the program.

Step 3  Investigate

Investigate the physical components and API code samples of the screen, the LED strip, the buttons, and the joystick.

Step 4  Modify and Make

Task 1: Modify the quiz and the answer in the example program.

Task 2: Modify the LED's lighting effects.

Tip:

Firefly: `cyberpi.led.play(name="firefly")`

Rainbow: `cyberpi.led.play(name="rainbow")`

Spoondrift: `cyberpi.led.play(name="spoondrift")`

Meteor Shower: `cyberpi.led.play(name="meteor_blue"); cyberpi.led.play(name="meteor_green")`

Flash: `cyberpi.led.play(name="flash_orange"); cyberpi.led.play(name="flash_red")`

Task 3: Use the joystick instead of the buttons to enter the answer.
Lesson 11  Data Protection and Passwords

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Features of CyberPi
Procedures

Step 1  Predict

Read the program.

```python
1. import cyberpi
2.
3. cyberpi.display.clear()
4. t = 0
5.
6. while True:
7.    print("Create a password")
8.    pin_1 = input("Type password: ")
9.    pin_2 = input("Type password again: ")
10.   if pin_2 == pin_1:
11.      print("Success!")
12.      break
13.  else:
14.      print("Passwords don't match. Try again.")
15.
16. print("Sign in your account")
17. cyberpi.console.println("Sign in")
18.
19. while t < 3:
20.    pin = input("Password: ")
21.    cyberpi.console.print("Password: ")
22.    cyberpi.console.println(pin)
23.    if pin == pin_1:
24.      cyberpi.console.println("Success!")
25.    break
26.  else:
27.    cyberpi.console.println("Incorrect. Try again.")
28.    t += 1
29.    if t == 3:
30.      cyberpi.console.println("Too many failed attempts.")
```
Questions:

- How to create a password for a new account?
- How to verify the password entered by a user?
- Compare the two `while` loops used in the example program. What is the difference between them?
- Identify the syntax that enables the following function: ‘A user is given 3 attempts to enter the password of the account.’ If the user fails 3 times, the user is blocked from entering the password.

Step 2  Run

Run the program.

Step 3  Investigate

Answer the questions.
Step 4   Modify

Task 1: Add the function that allows the user to create a username when the user signs in.

Task 2: Add some lighting effects as the indicator of a successful login. Use what you have learned to program the CyberPi.

   Tip:
   
   cyberpi.led.on(255, 255, 255)
   cyberpi.led.on(“green”, id = “all”)
   cyberpi.led.on(“red”, id = 3)
   cyberpi.led.show(“orange yellow cyan blue purple”)  
   cyberpi.led.play(name = “firefly”)  
   cyberpi.led.off(id = “3”)

Task 3: Modify the second ‘while’ loop. Use the LEDs as an indicator to remind the number of attempts.

Task 4: Modify the conditional expressions. Verify the following three conditions:

   The input username is incorrect;
   The input password is incorrect;
   Both the username and password are incorrect.
Step 5  Make

Develop a bank card reader

- First, create a PIN for the bank account and store it on the computer.
- When a sender starts a transaction, ask the sender to type the name (or other identification code) of the receiver and the amount of the transaction.
- Then ask the sender to type the PIN.
- Generate a random 4-digit verification code and display it on CyberPi’s screen. Ask the sender to enter the verification code.
- Check the PIN and verification code. If both are correct, display the transaction information (including the receiver’s name or ID and the transaction amount) on the screen. However, if either the PIN or the verification code is incorrect, ask the sender to insert them again. The sender has limited attempts (for example, 3 attempts).
- Ask the sender to check the transaction information and confirm the transaction by pressing Button B of the CyberPi.
Lesson 12  Normal Distribution

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

New Commands

cyberpi.display.set_brush()
cyberpi.barchart.add()
Features of CyberPi
Procedures

Step 1   Predict

Read the program.
1. `import` `cyberpi`, `random`
2. 
3. `cyberpi.display.clear()`
4. 
5. `n = int(input("The number of times to roll 2 dice: "))`
6. `t = 0`
7. `sum_list = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]`
8. `count_list = [0, 0, 0, 0, 0, 0, 0, 0, 0]`
9. 
10. `while` `t < n:`
11. `dice_x = random.randint(1, 6)`
12. `dice_y = random.randint(1, 6)`
13. `print("(", dice_x, ",", dice_y, ")")`
14. `result = dice_x + dice_y`
15. `t += 1`
16. `sum_index = sum_list.index(result)`
17. `count_list[sum_index] += 1`
18. 
19. `cyberpi.display.set_brush(128, 0, 0)`
20. `cyberpi.barchart.add(round(count_list[0]/n * 200, 2))`
21. `cyberpi.display.set_brush(220, 20, 60)`
22. `cyberpi.barchart.add(round(count_list[1]/n * 200, 2))`
23. `cyberpi.display.set_brush(255, 0, 0)`
24. `cyberpi.barchart.add(round(count_list[2]/n * 200, 2))`
25. `cyberpi.display.set_brush(205, 92, 92)`
26. `cyberpi.barchart.add(round(count_list[3]/n * 200, 2))`
27. `cyberpi.display.set_brush(233, 150, 122)`
28. `cyberpi.barchart.add(round(count_list[4]/n * 200, 2))`
29. `cyberpi.display.set_brush(255, 69, 0)`
30. `cyberpi.barchart.add(round(count_list[5]/n * 200, 2))`
31. `cyberpi.display.set_brush(255, 165, 0)`
32. `cyberpi.barchart.add(round(count_list[6]/n * 200, 2))`
33. `cyberpi.display.set_brush(255, 215, 0)`
34. `cyberpi.barchart.add(round(count_list[7]/n * 200, 2))`
35. `cyberpi.display.set_brush(240, 230, 140)`
36. `cyberpi.barchart.add(round(count_list[8]/n * 200, 2))`
37. `cyberpi.display.set_brush(255, 255, 0)`
38. `cyberpi.barchart.add(round(count_list[9]/n * 200, 2))`
39. `cyberpi.display.set_brush(154, 205, 50)`
40. `cyberpi.barchart.add(round(count_list[10]/n * 200, 2))`
Step 2  Run

Run the program.

Step 3  Investigate

Investigate the points below:

- The modules imported in this program;
- The variables displayed in the bar chart;
- Are ‘sum_list’ and ‘count_list’ variables? What are the values of them?
- The function that sets the colour of the bar chart;
- The function that creates the bars.

Step 4  Modify and Make

Task: Simulate the experiment of tossing three coins simultaneously. Calculate all the possible outcomes in this experiment and visualise the distribution with a bar chart.

<table>
<thead>
<tr>
<th>Head, Head, Head</th>
<th>🐷 🐷 🐷</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Lesson 13   Data Storage

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

New Commands

```python
import psutil

psutil.cpu_percent()
psutil.cpu_count()
psutil.cpu_freq()
psutil.virtual_memory()
cyberpi.linechart.add()
cyberpi.chart.set_name()
cyberpi.audio.play_tone()
```
Features of CyberPi
Procedures

Step 1   Predict

Read the program.

```python
1. import cyberpi
2. import psutil
3.
4. cyberpi.chart.clear()
5. while True:
6.     CPU = psutil.cpu_percent()
7.     mem = psutil.virtual_memory()
8.     mem_p = mem.percent
9.     cyberpi.display.set_brush(0, 0, 255)
10.    cyberpi.linechart.add(int(CPU))
11.    cyberpi.display.set_brush(255, 255, 0)
12.    cyberpi.linechart.add(int(mem_p))
13.    print("CPU:", CPU, "% Memory:", mem_p, ")
```

Step 2   Run

Run the program.

Step 3   Investigate

Investigate the points below:

- The library for calling functions to monitor the CPU and memory usage
- The two variables plotted in the line chart;
- The function that plots the lines;
- The function that set the color of the line.
Step 4 Modify and Make

Task: Add control structures and light effects to the example program. Define the thresholds of the alarm and the corresponding alarm indicators.

Note: CPU performance

<table>
<thead>
<tr>
<th>Lower CPU performance</th>
<th>Higher CPU performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-core</td>
<td>Multi-core</td>
</tr>
<tr>
<td>Low clock speed</td>
<td>High clock speed</td>
</tr>
<tr>
<td>Small or no cache</td>
<td>Large, multi-level cache</td>
</tr>
</tbody>
</table>
Lesson 14  Remix Culture

Worksheet

K-W-L Chart

<table>
<thead>
<tr>
<th>What I Know</th>
<th>What I Wonder</th>
<th>What I Learned</th>
</tr>
</thead>
</table>

New Commands

```python
from pynput.keyboard import Key, Listener
from pynput.keyboard import Button, Controller
cyberpi.is_tiltleft
cyberpi.is_tiltright
cyberpi.is_tiltforward
cyberpi.is_tiltback
cyberpi.audio.play_music()
```
Features of CyberPi

- Light sensor
- Microphone
- Electronic module interface
- Button A (Return key)
- HOME button (Enter CyberOS)
- Power & data interface (Type-C)
- WiFi + Bluetooth ESP32
- Full-color display
- Button B (Confirmation key)
- Gyroscope
- Accelerometer
- RGB LED (Five)
- Speaker
Procedures

Step 1    Predict

Read the program.
1. import cyberpi
2. from pynput.keyboard import Key, Listener
3. cyberpi.audio.set_vol(50)

4. def bar1():
   5.     cyberpi.audio.play_music(60, 0.2)
   6.     cyberpi.audio.play_music(64, 0.2)
   7.     cyberpi.audio.play_music(67, 0.2)

8. def bar2():
   9.     cyberpi.audio.play_music(64, 0.2)
  10.     cyberpi.audio.play_music(65, 0.2)
  11.     cyberpi.audio.play_music(69, 0.2)

12. def bar3():
  13.     cyberpi.audio.play_music(64, 0.2)
  14.     cyberpi.audio.play_music(67, 0.2)
  15.     cyberpi.audio.play_music(71, 0.2)

16. def bar4():
  17.     cyberpi.audio.play_music(65, 0.2)
  18.     cyberpi.audio.play_music(69, 0.2)
  19.     cyberpi.audio.play_music(72, 0.2)

20. def on_press(key):
  21.     if key.char == "1":
  22.         cyberpi.led.on("red")
  23.         bar1()
  24.     if key.char == "2":
  25.         cyberpi.led.on("orange")
  26.         bar2()
  27.     if key.char == "3":
  28.         cyberpi.led.on("yellow")
  29.         bar3()
  30.     if key.char == "4":
  31.         cyberpi.led.on("green")
  32.         bar4()

33. def on_release(key):
  34.     cyberpi.led.off()
  35.     pass

36. with Listener(on_press=on_press, on_release=on_release) as listener:
  37.     listener.join()
Step 2  Run

Run the program.

Step 3  Investigate

Investigate the points below:

- Identify the new module that can monitor the input from your keyboard.
- What is meant by the Python keyword ‘def’?
- Why does it separate the set of expressions within each ‘def’ code block?
- How to produce interactive sound and light effects?

Step 4  Modify and Make

Task 1: Modify the parameters in the example program. Use letters to replace the numeric parameters.

Task 2: Modify the ‘bar’ functions. Find some pieces of songs from music textbooks and combine different parts of them together to make a new song. Consider how to combine the sound effects and LED lights.