PLS Practice Worksheet

Programming Language Subset Skills Builder

Purpose: This worksheet helps you practice using the Programming Language Subset (PLS) document effectively during Paper 2 examinations.

Instructions: For each question, identify the PLS section you would consult, then write the correct Python code. Time yourself - aim to find each reference within 20 seconds.

Papers Referenced: June 2022, June 2023, June 2024 (Edexcel 1CP2/02)

Name: _____ Date: _____

Class: _____ Score: ___ / 50

Section A: PLS Scavenger Hunt (10 marks)

Find the following in your PLS as quickly as possible. Write the page number and the exact syntax.

1. How do you check if a string contains only digits?

Page: ___ Syntax: _____

2. What is the operator for integer division?

Page: ___ Syntax: _____

3. How do you add an item to the end of a list?

Page: ___ Syntax: _____

4. How do you open a file for appending?

Page: ___ Syntax: _____

5. How do you generate a random integer between 1 and 100?

Page: ___ Syntax: _____

6. What is the syntax for a while loop?

Page: ___ Syntax: _____

7. How do you round a number to 3 decimal places?

Page: ___ Syntax: _____

8. How do you convert a character to its ASCII value?

Page: ___ Syntax: _____

9. How do you get the smallest integer not less than a value?

Page: ___ Syntax: _____

10. How do you split a string by commas?

Page: ___ Syntax: _____

Section B: Error Identification (10 marks)

For each error below, identify: (a) the type of error, (b) the PLS page to check, (c) the corrected code.

1. Context: Boolean initialisation

```
found = false
```

Error type: _____ PLS Page: ___

Corrected code: _____

2. Context: Integer assignment

```
wavelength = 0123
```

Error type: _____ PLS Page: ___

Corrected code: _____

3. Context: Library import

```
import randum
```

Error type: _____ PLS Page: ___

Corrected code: _____

4. Context: Iteration syntax

```
for item in myList
```

Error type: _____ PLS Page: ___

Corrected code: _____

5. Context: Logic for 'outside range'

```
if (x < 5) and (x > 10):
```

Error type: _____ PLS Page: ___

Corrected code: _____

6. Context: Random number (complete the line for 0-9)

```
index = random.
```

Error type: _____ PLS Page: ___

Corrected code: _____

7. Context: Type conversion

```
print(int("hello"))
```

Error type: _____ PLS Page: ___

Corrected code: _____

8. Context: Array indexing

```
name = exerciseTable[index + 1] # index is 0-4, array has 5 items
```

Error type: _____ PLS Page: ___

Corrected code: _____

9. Context: Range behaviour

```
range(numItems - 1) # want exactly numItems iterations
```

Error type: _____ PLS Page: ___

Corrected code: _____

10. Context: File mode string

```
file = open("data.txt", r)
```

Error type: _____ PLS Page: ___

Corrected code: _____

Section C: Code Completion (15 marks)

Complete each code snippet using the PLS. Write the PLS page reference you used.

1. File Reading and Processing (3 marks)

Read data from a CSV file and split each line:

```
# Open file for reading dataFile = _____("students.txt", ____) # Read each line
for line in dataFile: # Remove whitespace from ends line = line._____() # Split by
comma parts = line._____(",") dataFile._____()
```

PLS Pages used: _____

2. Input Validation (3 marks)

Accept a price (must be positive decimal) and quantity (must be positive whole number):

```
# Get price as a decimal number price = _____(input("Enter price: ")) # Get
quantity as a whole number quantity = _____(input("Enter quantity: ")) # Check
both are positive (greater than zero) if price ____ 0 _____ quantity ____ 0: total =
price * quantity print("Valid order") else: print("Invalid input")
```

PLS Pages used: _____

3. Random Selection from Array (3 marks)

Select a random item from an array:

```
import _____ colours = ["red", "blue", "green", "yellow", "purple"] # Generate
random index (0 to 4 inclusive) index = random._____(0, _____(colours) - 1)
# Get the colour at that index selectedColour = colours[_____] print("Selected:",
selectedColour)
```

PLS Pages used: _____

4. String Key Generation (3 marks)

Create a key from first 2 letters of name + ID divided by 100:

```
name = "Johnson" studentID = 45678 # Get first two letters of name firstTwo =
name[____:____] # Integer divide ID by 100 idPart = studentID _____ 100 # Combine into
key (convert number to string) key = firstTwo + _____(idPart) # Result should be "Jo456"
```

PLS Pages used: _____

5. Function with Return Value (3 marks)

Complete the function that calculates and returns area:

```
import math _____ calculateCircleArea(radius): # Area = pi * r squared area = math._____
* radius _____ 2 _____ area # Call function and display rounded result result =
calculateCircleArea(5) print("Area:", _____(result, 2))
```

PLS Pages used: _____

Section D: Problem Solving with PLS (15 marks)

For each scenario, list the PLS sections you would need, then write the complete solution.

1. Ingredient Calculator (5 marks)

A bakery needs to calculate how many bags of flour to order.
- Each cake requires 0.4 bags of flour
- Bags cannot be ordered as fractions (must round UP)
- Accept the number of cakes from user input
- Display the partial bags needed and whole bags to order

PLS sections needed: _____

Solution:

_____

_____

_____

_____

_____

_____

_____

_____

2. Menu-Driven Program (5 marks)

Create a simple menu system:
- Option 1: Display a random number between 1 and 100
- Option 2: Display "Hello World"
- Option 0: Exit the program
- Loop until user chooses to exit
- Tell user if they enter an invalid option

PLS sections needed: _____

Solution:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

3. Date Validation (5 marks)

Validate a date of birth entered as a string (format: ddmmyyyy):
- Check that the string contains only digits
- Check that the string is exactly 8 characters long
- If valid, calculate the sum of ASCII values of all characters
- Display appropriate messages for valid/invalid input

PLS sections needed: _____

Solution:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

ANSWER KEY (Teacher Copy)

Section A: Scavenger Hunt

1. Page 11: .isdigit()
2. Page 6: //
3. Page 10: .append()
4. Page 8: open(, "a")
5. Page 13: random.randint(1, 100)
6. Page 7: while :
7. Page 9: round(, 3)
8. Page 9: ord()
9. Page 13: math.ceil()
10. Page 11: .split(",")

Section B: Error Identification

1. NameError, Page 5: found = False
2. SyntaxError, Page 5: wavelength = 123 (no leading zero)
3. SyntaxError, Page 13: import random
4. SyntaxError, Page 7: for item in myList: (missing colon)
5. Logic, Page 6: if (x < 5) or (x > 10): (use OR not AND)
6. Incomplete, Page 13: index = random.randint(0, 9)
7. ValueError, Page 5: Cannot convert non-numeric string
8. IndexError, Page 5: Remove +1, use exerciseTable[index]
9. Logic, Page 9: range(numItems) - range excludes endpoint
10. SyntaxError, Page 8: open("data.txt", "r") - quotes around mode

Section C: Code Completion

1. open, "r", strip(), split, close() - Pages 8, 11
2. float, int, >, and, > - Pages 5, 6
3. random, randint, len, index - Pages 9, 13
4. [0:2], //, str - Pages 6, 9, 12
5. def, pi, **, return, round - Pages 8, 9, 13

Section D: Problem Solving - Key Points

1. Ingredient Calculator: import math, float(input()), math.ceil() - Pages 5, 8, 13

2. Menu System: while loop, if/elif/else, random.randint(), int(input()) - Pages 5, 7, 8, 13

3. Date Validation: isdigit(), len(), ord(), for loop - Pages 7, 9, 11