Zig Zag Education

Computing Level 3 AAQ | Pearson BTEC | 610/3963/9

Worksheets for BTEC National in Computing

Unit 1: Programming Fundamentals

Download support files from zzed.uk/productsupport

zigzageducation.co.uk

POD 12999

Contents

Teacher's Introduction

This resource is designed to support the teaching of **Unit 1** in the **BTEC Computing (AAQ)** qualification. It is intended for **first teaching from September 2025**, with the **first external assessment taking place in 2026**.

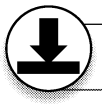
Unit 1 is externally assessed through a **single written examination**, which is worth **90 marks** and lasts for **2 hours and 30 minutes**.

Included in this resource are **31 worksheets**, comprising **89 programming challenges**, covering practice tasks or supporting theory. Each set of programming challenges builds on the last, creating a structured, progressive path through the core programming concepts. This approach supports **knowledge recall** and encourages the development of **new programming skills**. The design ensures students revisit, and reinforce, previous learning as they move forward in the course.

Worksheets are designed for students to write their answers in their exercise books or on separate paper. Write-on versions of pages where this may be useful are provided in an appendix at the end of the resource.

All worksheets and programming challenges come with **full solutions**, making them ideal for use as **in-class activities** or **homework tasks**. They are specifically designed to **consolidate learning** and enhance student confidence and competence in programming and theory.

January 2026



A code file, **student_reports.py**, has been provided for worksheets A3.4 and A3.5 at **zzed.uk/ps**

Lesson Overview

Spec. reference	Worksheet focus	Worksheet aim	Worksheet/s	Programming challenges	Write-on appendix provided
A1.1	Number systems	Performing conversions between number systems	✓		✓
A1.2	Negative numbers and floating-point numbers	Using binary to represent negative and floating-point numbers	✓		✓
A1.3	Binary arithmetic	Performing binary arithmetic (addition, subtraction, multiplication, division)	✓		✓
A1.4	Binary shifting	Shifting binary numbers	✓		✓
A1.5	Overflow errors	Understanding overflow errors and how to manage them	✓		✓
A2.1 to A2.3	Python operators	Using mathematical, relational, and Boolean operators in Python	✓		✓
A2.4 and A2.5	Variables and data types	Declaring, initialising, and using constants and variables to solve simple problems	✓		✓
A3.1	Sequencing	Arranging program actions logically to solve problems		✓	
A3.2	Selection	Using IF/ELIF/ELSE and switch/case statements		✓	
A3.3	Iteration	Repeating actions using loops to improve accuracy and reliability		✓	

Continued overleaf

Lesson Overview					
Spec. reference	**Worksheet focus**	**Worksheet aim**	**Worksheet/s**	**Programming challenges**	**Write-on appendix provided**
A3.4 and A3.5	User-defined functions	Creating functions, using local/global variables, and modularising code ↓The code file, **student_reports.py**, has been provided for these worksheets at **zzed.uk/ps**		✓	✓ (A3.5)
A3.6	Recursion	Using recursion to solve problems		✓	
B1.1 and B2.4	One-dimensional and two-dimensional arrays	Structuring data with arrays and using built-in methods when writing program code		✓	
B1.2	Data structures	Using tuples and dictionaries to solve problems in python		✓	
B1.3	Stacks and queues	Storing and accessing data using stacks and queues (FIFO, LILO)	✓		✓
B2.1 and B2.2	Input and output numerical functions	Understanding Python's built-in functions for numerical data	✓		✓
B2.3	String manipulation	Using built-in string handling functions	✓		✓
B2.5	External text files	Reading from and writing to external text files	✓		✓
C1.1 to C1.3	Computational thinking and algorithms	Applying logic (pattern recognition, decomposition, abstraction) to solve problems	✓		✓
C1.4	Producing algorithms	Representing algorithms visually using flowcharts	✓		
C1.5	Interpreting algorithms	Using dry runs and trace tables to interpret algorithms	✓		✓
C1.6	Common searching and sorting algorithms	Understanding common search/sort algorithms and evaluating their use	✓		✓
C2.4	Testing	Performing effective testing of program code	✓		✓
D1.1	Use of third-party code	Exploring issues related to using others' code	✓		
D1.2	Diversity and inclusion	Understanding the importance of inclusive software design and preventing bias in datasets	✓		
D1.3	Common challenges in software development	Mitigating common development challenges	✓		

Topic A1 Number systen

Worksheet A1.1 Conversions of number systems

Conversion of numbers between number systems.

1. Copy and complete the table below showing the value of each digit in th

Denary number	2	6	9	
Power of 10		10^4		
Value of power	100000		1000	
Calculation of digit's value		Six lots of 10000		
Total value of digit		60000		

2. Copy and complete the table below showing the value of each digit in th

Binary number	1	1	0	1	
Power of 2		2^5		2^3	
Value of power	64				
Calculation of digit's value			No lots of 16		
Total value of digit	64				

3. Convert the following numbers.
   a) Convert **13** to binary
   b) Convert **45** to binary
   c) Convert **102** to binary
   d) Convert **2..** to binary
   e) Convert **73** to binary
   f) Convert **1010** to d
   g) Convert **11001** to
   h) Convert **1111111**
   i) Convert **100000** to
   j) Convert **101101** to

Worksheet A1.2 Use of binary to represent negative and

Use of binary to represent negative and floating-point numbers

1. Convert the following 8-bit two's complement binary numbers into decim
   a) 11110011 b) 10010101 c) 11111111

2. Convert these denary negative numbers into 8-bit two's complement.
   a) -13 b) -1 c) -104

3. Convert the following 8-bit fixed-point binary numbers into denary.
   a) 0011.0000 b) 0101.1000 c) 0101.1000

4. Below are 8-bit floating-point numbers with the following format: 1 bit f
   4 bits mantissa. Interpret their decimal value.
   a) 0 1010 b) 1 011 0100 c) 0 110 0000

5. Convert the following denary numbers into 8-bit floating point using the
   exponent, and 4 bits for mantissa.
   a) 3.5 b) -1.25 c) 0.5

Worksheet A1.3 Performing addition, subtraction, divisio

Binary arithmetic (addition, subtraction, multiplication and divis

Perform the following binary additions and convert your answers to denary.

1. 00101101 + 00010111
2. 01100011 + 00011100
3. (illegible)

Perform the following binary subtractions and convert your answers to denary

4. 01011010 − 00010101
5. (illegible) − 00011000
6. (illegible)

Perform the following binary multiplications and convert your answers to den

7. 00000(illegible)
8. 00000110 × 00000010
9. (illegible)

Perform the following binary divisions and convert your answers to denary.

10. 00110000 ÷ 00000010
11. 00010100 ÷ 00000010
12. (illegible)

13. What would it mean when performing subtraction, and calculation produ leading 1 in two's complement?

Worksheet A1.4 Binary shifts

Shift the binary numbers to the left as indicated. Then convert to denary.

1. 00001101 << 1
2. 00001101 << 2
3. 00010010 << 3
4. (illegible)

Shift the binary numbers to the right as indicated. Then convert to denary.

5. 01110000 >> 1
6. 00011000 >> 2
7. (illegible)0 000 >> 3
8. (illegible)

9. What is the effect of a 1-bit left shift on a binary number?

10. What is the effect of a 2-bit right shift on a binary number?

11. Explain how left shifting can be used to multiply a number by powers of 2

12. The 8-bit binary number 00110100 is shifted left by 2 bits.
    a) What happens to the bits that "fall off" the left?
    b) What should you do with the empty spaces on the right?

Worksheet A1.5 Overflow errors

1. What is an overflow error?

2. Do these binary additions cause overflow in 8 bits?
    a) 11111111 + 00000001 → YES / NO
    b) 10000000 + 10000000 → YES / NO
    c) 00001111 + 00000001 → YES / NO
    d) 01111111 + 00000001 → YES / NO

3. List two problems an overflow error could cause in real life.

4. What are two ways to prevent or handle overflow?

5. Imagine a video game score is stored as 8-bit.
   What happens if your score goes from 255 to 256?

Topic A2 Fundamentals of data

Worksheets A2.1 to A2.3 Operators

Relational operators compare values and return True or False.

1. Identify the correct relational operator so that the output is True.
   a) 7 ___ 7 → True b) 4 ___ 7 → True c) 9 ___ 5 → True

2. What will each of the following expressions return?
   a) 5 == 5 b) 10 < 3 c) 8 >= 8

Boolean operators combine multiple conditions.

3. Give the correct Boolean operator to go in the blanks.
   a) True ___ False → ___ c) ___ True → ___
   b) False ___ False → ___ d) (5 > 3) ___ (2 == 2) →

4. Evaluate the following and write the output.
   a) (6 > 2) and (3 < 5) c) (7 < 2) or (8 == 8)

5. Identify the correct arithmetic operator.
   a) 10 ___ 3 gives → 1 c) 3 ___ 2 equals 2 → 1
   b) 7 ___ 2 gives → 3 d) 3 ___ 5 equals 15 →

6. Write the result of the following expressions.
   a) 8 + 4 * 2 b) 9 % 4 c) 5 // 4

Worksheets A2.4 and A2.5 Variables

Data types used in programming (string, integer, float/real, Boolean). Declaration, initialisation, and use of constants and variables.

1. Match each data type to the correct description.

	Data type
a)	String
b)	Integer
c)	Float
d)	Boolean

	Description
1)	A number with a decimal point
2)	A value that can only be True o
3)	A group of characters or text (e.g
4)	A whole number (e.g. 7, -3)

2. Choose the correct data type for each value.
   a) "123" → String / Integer / Boolean c) 7.0 → Float / Integ
   b) True → String / Boolean / Float d) "False" → Boolean

3. Give the correct data type to go in the blanks.
   a) "Hello world" is a __. c) False is a __ value.
   b) 25 is an example of an __. d) 3.1415 is a __.

4. Identify the missing words.
   a) A variable is a named space in memory that can __.
   b) A constant is a named value that __ change during a program.
   c) The process of giving a variable a value when it is created is called __.
   d) Declaring a variable means __.

5. Write down what each line of code is doing.
   a) name = "Alice" b) PI = 3.14 c) score = 0

Worksheet A3.1 The use of appropriate sequencing in pr

Programming challenges 1 to 10

Challenge 1
Write a program that asks the user to enter two numbers. The program should
of the two numbers.

Challenge 2
The program asks the user to enter a temperature in Celsius. The program sho
equivalent temperature in Fahrenheit using the formula: F = (C * 9/5) + 32.

Challenge 3
Design a program that asks the user to enter a number. The program should c
of the number.

Challenge 4
Write a program that asks the user to enter the length and width of a rectang
calculate and output the area of the rectangle using the formula: area = leng

Challenge 5
The program asks the user to enter the radius of a circle. The program should
circumference of the circle using the formula: circumference = 2 * pi * radius (

Challenge 6
Design a program that asks the user to enter their age. The program should c
months (assuming 1 year = 12 months).

Challenge 7
Write a program that asks the user to enter a speed in kilometres per hour (k
calculate and output the equivalent speed in miles per hour (mph) using the
1 km/h = 0.621371 mph.

Challenge 8
The program asks the user to enter a length in inches. The program should ca
equivalent length in centimetres using the conversion factor: 1 inch = 2.54 ce

Challenge 9
Design a program that asks the user to enter the base and height of a triangle
and output the area of the triangle using the formula: area = (base * height) /

Challenge 10
Write a program that asks the user to enter their weight in kilograms and thei
program should calculate and output their body mass index (BMI) using the fo
BMI = weight / (height * height).

Programming challenges 11 to 20

Challenge 11
Write a program that asks the user to input a number. If the number is even, t
"The number is even." If it is odd, it should print "The n er is odd."

Challenge 12
Write a program that asks th er t input a year. The program should check
print "Leap year" if it is a leap year" otherwise. (A leap year is divisible
also divisi 40

Challenge 13
Write a program that asks the user to input a grade between 0 and 100. The p
if the grade is greater than or equal to 90, "Good" if it is between 80 and 89, "A
79, and "Fail" if it is below 70.

Challenge 14
Write a program that asks the user to input two numbers. The program shoul
largest and smallest number.

Challenge 15
Write a program that asks the user to input a character. The program should c
lowercase letter (a–z) and print "Lowercase letter" if it is, o "Not a lowercase
Hint: use the .islower( ) or .isupper( ) method.

Challenge 16
Write a program that as to input three numbers. The program shou
the numbe fo d triangle. If the sum of any two sides is greater t
triangle; ot e, it is not.

Challenge 17
Write a program that asks the user to input a number. The program should ch
negative, or zero, and print the corresponding message.

Challenge 18
Write a program that asks the user to input a number. The program should ch
square and print "Perfect square" if it is, or "Not a perfect square" otherwise.
(A perfect square is a number that is the square of an integer.)

Challenge 19
Write a program that asks the user to input three n rs. The program shou
largest and smallest numbers among them

Challenge 20
Write a progr th e user to input a letter. The program should check i
or a conso it is a vowel, it should print "Vowel"; if it is a consonant, it s

Programming challenges 21 to 30

Challenge 21
Ask the user to enter a number (1–7). Use a match statement to print the cor

Challenge 22
Ask the user to enter a traffic light co[illegible], yellow, or green). Use match to
(e.g. "Stop", "Slow down", "Go")

Challenge
Ask the use[illegible]oose an operation: add, subtract, multiply, divide. Then ask f
perform the correct calculation.

Challenge 24
Ask the user to enter an animal (dog, cat, cow, duck). Use match to print the s

Challenge 25
Ask the user to enter a letter grade (A, B, C, D, F). Use a match-case statement
(e.g. "Excellent!", "Good job!", "You're doing fine", "You must try harder", "Invalid

Challenge 26
Ask the user to choose a language (English, Spanish, French, German). Use mat
that language.

Challenge 27
Ask the user to enter a planet na[illegible]y to Neptune). Use match to print

Challenge
Show a sh[illegible] menu (e.g. 1 = Burger, 2 = Pizza, 3 = Salad). Ask the user to
to print wha[illegible]ey ordered.

Challenge 29
Simulate rolling a die. Generate a number from 1–6 and use match to print a fu

Challenge 30
Ask the user to enter a mood (happy, sad, angry, excited). Use match to print a
the emotion.

Programming challenges 31 to 40

Challenge 31
Write a program that prints the numbers from 1 to 10 using a loop.

Challenge 32
Write a program that calculates and prints the sum of all even numbers from

Challenge 33
Write a program that asks the user to enter a positive integer. The program sh
factorial of the number using a loop.

Challenge 34
Write a program that prints the multiplication table of a given number. The p
input a number and then print its multiplication table from 1 to 10 using a lo

Challenge 35
Write a program that prints the Fibonacci series up to a given number. The pr
input a number and then print the Fibonacci series up to that number using a

Challenge 36
Write a program that asks the user to input a sentence. The program should c
vowels (a, e, i, o, u) in the sentence using a loop.

Challenge 37
Write a program that generates and prints a random number between 1 and 100 (
you.) The program should keep generating numbers until it generates a number t

```
import random
random_num = random.randint(1, 100)
```

Challenge 38
Write a program that prints the prime numbers between 1 and 100 using a lo
than 1 and has no divisors other than 1 and itself.)

Challenge 39
Write a program that asks the user to enter a password. The program should k
until the user enters the correct password using a loop.

Challenge 40
Write a program that asks the user to enter a positive integer. The program sh
sum of all the digits in the number using a loop.

Worksheet A3.4 User defined functions

Programming challenges 41 to 50

Challenge 41
Write a program that defines a function greet_user() which prints "Hello, user!" to display the greeting.

Challenge 42
Write a program that defines a function calculate_area() which takes two parameters. The function should calculate and return the area of a rectangle (area = length * width). Prompt the user for values for length and width, call the function, and print the calculated area.

Challenge 43
Write a program that defines a function is_even() which takes an integer as a parameter. The function should return True if the integer is even, and False otherwise. Prompt the user to input a number, call the function, and print whether the number is even or odd.

Challenge 44
Write a program that defines a function calculate_factorial() which takes an integer as a parameter. The function should calculate and return the factorial of the parameter. Prompt the user to input a number, call the function, and print the factorial.

Challenge 45
Write a program that defines a function is_prime() which takes an integer parameter. The function should return True if the parameter is a prime number, and False otherwise. Prompt the user to input a number, call the function, and print whether it is prime or not.

Challenge 46
Write a program that defines a function gcd() which takes two integer parameters a and b. The function should calculate and return the Greatest Common Divisor (GCD) of a and b. Prompt the user to input two numbers, call the function, and print the GCD.

Challenge 47
Write a program that defines a function check_temperature() which takes a temperature as a parameter. The function should print a message based on the value: Below 0 "Freezing", between 0 and 30 "Warm", and above 30 "Hot". Prompt the user t input a temperature, call the function, and print the message.

Challenge 48
Write a program that defines a function sum_of_squares() which takes an integer parameter n. The function should calculate and return the sum of squares of all integers from 1 to n. Prompt the user to input a number, call the function, and print the result.

Challenge 49
Write a program that defines a function lcm() which takes two integer parameters a and b. The function should calculate and return the Least Common Multiple (LCM) of a and b. Prompt the user to input two numbers, call the function, and print the LCM.

Challenge 50
Write a program that defines a recursive function fibonacci() which takes an integer parameter n. The function should return the nth number in the Fibonacci sequence, where Fibonacci(0) = 0, Fibonacci(1) = 1, and each subsequent number is the sum of the previous two. Prompt the user to input a number, call the function, and print the Fibonacci number at that position.

Worksheet A3.5 Local and global variables

User defined functions. Local and global variables.

Using the program ***student_reports.py***, complete the following tasks.

1. Identify all user-defined **function names**, including the line number they
2. State the purpose of each user-defined **function**.
3. Identify all **variables** in the user-defined functions, including the line n
4. Identify all **arguments** in the user-defined functions, including the line n
5. Identify all **global variables**, including the line number they appear on.
6. Identify all **local variables**, including the line number they first appear on

student_reports.py

```
1  def calculate_average(scores):
2      total = sum(scores)
3      return total / len(scores)
4
5  def convert_to_grade(average):
6      if average >= 90:
7          return "A"
8      elif average >= 80:
9          return "B"
10     elif average >= 70:
11         return "C"
12     elif average >= 60:
13         return "D"
14     else:
15         return "F"
16
17 def print_report(name, scores):
18     average = calculate_average(scores)
19     grade = convert_to_grade(average)
20     print("----- Student Report -----")
21     print("Name:", name)
22     print("Scores:", scores)
23     print("Average Score:", round(average, 2))
24     print("Final Grade:", grade)
25     print("--------------------------")
26
27 student_name = input("Enter student's name: ")
28 student_scores = []
29
30 for i in range(3):
31     score = float(input(f"Enter score {i + 1}: "))
32     student_scores.append(score)
33
34 print_report(student_name, student_scores)
```

Programming challenges 51 to 60

Challenge 51
Write a recursive function to calculate the factorial of a number n.

Challenge 52
Write a recursive function to return the nth number in the Fibonacci sequence

Challenge 53
Use recursion to sum the digits of a number.

Challenge 54
Write a function to reverse a string using recursion.

Challenge 55
Print a countdown from a given number to 1 using recursion.

Challenge 56
Use recursion to check if a string is a palindrome (reads the same forwards and

Challenge 57
Given a list of numbers, recursively calculate the sum of all elements.

Challenge 58
Write a function `power(base, exponent)` that uses recursion to compute

Challenge 59
Recursively print all elements in a list, including those inside nested sublists.

Challenge 60
Use Euclid's algorithm to recursively find the GCD of two numbers.

Worksheet B1.1a One-dimensional data structures

Programming challenges 61 to 70

Challenge 61
Write a program that takes a list of 10 numbers as input and calculates the sum of

Challenge 62
Write a program that takes a list of 10 words as input and prints the length of

Challenge 63
Write a program that combines two lists (of your choice) into a single list, alternating between each list. The two lists you choose must be the same size.

Challenge 64
Write a program that takes a list of numbers as input and finds the largest number. The user should decide how many items to input.

Challenge 65
Write a program that takes a list of strings as input and prints all the strings that are longer than 5. The user should decide how many items to input.

Challenge 66
Write a program that takes a list of numbers as input and returns a new list with duplicates removed (removing duplicates). The user should decide how many items to input.

Challenge 67
Write a program that takes a list of names as input and sorts them in alphabetical order. The user should decide how many items to input.

Challenge 68
Write a program that takes a list of numbers as input and calculates the product of all the numbers.

Challenge 69
Write a program that asks the user to enter a sentence. The program should count and display the number of words in the sentence.

Challenge 70
Write a program that takes a list of numbers as input and calculates the average.

Worksheet B1.1b Two-dimensional data structures

Programming challenges 71 to 80

Challenge 71
Write a program that takes a 2D list (a list of lists) of integers and calculates t
the 2D list.

Challenge 72
Write a program that takes a 2D list of strings and prints the length of each st

Challenge 73
Write a program that merges two 2D lists of equal dimensions by alternating
correspond sition.

Challenge 74
Write a program that takes a 2D list of numbers and finds the largest numbe

Challenge 75
Write a program that takes a 2D list of strings and prints only the strings wit

Challenge 76
Write a program that takes a 2D list of numbers and returns a flattened list w
(removing duplicates).

Challenge 77
Write a program that takes a 2D list of names and sorts each inner list alpha

Challenge 78
Write a program that takes a 2D list of integers and calculates the product o

Challenge 79
Write a program that take list where each inner list is a sentence split
total number of wo sentences.

Challenge
Write a program that takes a 2D list of numbers and calculates the average va

Worksheet B1.2a Python dictionaries

Challenge 81
Create a program that takes a string input from the user and counts the frequ
Display the word frequency using a dictionary where the keys are the unique
counts of each word.

Challenge 82
Develop a phone book program that allows users to add contacts, remove conta
all contacts. Implement this using a dictionary where the keys are names, and th

Challenge 83
Build an inventory management system for a store. The program should allow
inventory, remove items, update quantities, and display the current inventory.
items as keys and their quantities as values.

Challenge
Create a program to manage student grades. Allow users to input student na
grades. Implement functionalities to add new students, remove students, upd
student names along with their grades. Store this information using a diction
and grade lists as values.

Worksheet B1.2b Python tuples

Challenge 85
Create a tuple named days_of_week that stores all seven days of the week.
Then print the first and last day.

Challenge 86
Create a tuple called person that stores a name, age and city.
Then print each piece of information separately.

Challenge 87
Create a dictionary where the keys are city names and the values are tuples of (latitude, longitude).
Then allow the user to input a city name and return its coordinates.

Challenge 88
Create a list of student records where each student is stored as a tuple: (name
Then loop through the list and print out a sentence for each student.

Challenge 89
Write a function min_and_max(numbers) that returns a tuple with the smallest list of numbers.

Worksheet B1.3 Stacks and queues

1. A stack is a data structure that follows the __ rule.
2. A queue is a data structure that follows the __ rule.
3. In a stack, the most recently added item is the __ one to be removed.
4. In a queue, the item added first is the one removed __.
5. Which data structure would be more useful in the scenarios below?

Scenario	Stack or queue
Printer job list	
Undo feature in a text editor	
Customers lining up at a checkout	
Website back-button history	

6. Describe **one reason** why a programmer would use a stack instead of a queue.
7. Given the following actions, state the final contents of the stack and queue
   - Add A
   - Add B
   - Add C
   - Remove one item

Worksheets B2.1 and B2.2: Input and output; Numerical

1. To use the random module, you must first write:
   ___ `random`

2. `random.randint(1, 5)` could return which of the following?
   a) 1 b) 3.5 c) 6 d) 5

3. What does `random.choice(['red', 'blue', 'green'])` do?

4. What does `random.shuffle(my_list)` do?

5. Write code to do the following.
   a) Print a random number between 10 and 20
   b) Shuffle the list ['apple', 'banana', 'cherry'] and print the result
   c) Pick a random letter from the word "coding"

6. Match the function to its output.

a) round(7.6)	
b) int(7.6)	
c) round(3.5)	
d) round(2.5)	

1)	7
2)	2
3)	8
4)	4

7. Predict the output.
   a) int(9.99) b) round(2.5) c) math.floor(-3.2)

8. What is the difference between truncating and rounding?

Worksheet B2.3: String handling

1. What will the following code output?

```
first = "Good"
second = "Morning"
message = first + " " + second
print(message)
```

2. Identify the missing word to join the three words.

```
word1 = "Computer"
word2 = "Scien
word3 = "R
sente = ___
print(sentence)
```

3. What does .split() do? Select the correct answer.
   a) Joins strings together
   b) Removes punctuation
   c) Breaks a string into a list of words

4. What will this code output?

```
sentence = "I love Python"
words = sentence.split()
print(words)
```

5. Turn this string into a list using a built-in function.

```
shopping = "milk,bread,eggs"
```

6. Complete the code to clean up the user's name.

```
name = input("Enter your name: ")
clean_name = name._____().______()
print(f"Hello, {clean_name}!")
```

7. Match the method to its use.

Method		Use	
a)	`.strip()`	1)	Capitalises first letter of each w
b)	`.lower()`	2)	Makes everything lowercase
c)	`.title()`	3)	Removes extra spaces from inpu

8. Fill in the missing line to print name and age clearly using a formatted s

```
name = "Zara"
age = 17
print(____)
# Expected: Zara is 17 years old.
```

9. True or False?

   `f"{name} is {age} years old"` is an example of a formatted str

10. Write a program that:
    - Takes a full sentence as input
    - Splits it into words
    - Counts, and outputs, how many words
    - Prints each word on a new line

Worksheet B2.5: External text files

1. Define the term delimiter in the context of text files.

2. Underline the delimiter in each example below.
   a) Alice,17,Maths
   b) John 88 Physics
   c) 101|Bobby|History

3. Draw each of the following delimiters.
   a) Co
   b) Pi

4. Write one line of code to open a file called data.txt in write mode.

5. Complete this code to write three lines into a file called greetings.txt

```
with open("greetings.txt", "w") as f:
    f.write("Hello\n")
```

6. Given this code, what will be printed?

```
with open("scores.txt", "r") as f:
    for line in f:
        name, score = line.strip().split(",")
        print(name)
```

scores.txt contents:

```
Asha,15
Brian,14
...
```

7. List two benefits of using .txt files to store program data.

8. List two drawbacks of using .txt files.

9. Would a .txt file be a good choice for storing users' login details in a web

10. Write a program that reads a file called students.txt where:
    - each line is in the format name,grade
    - prints only the names of students who got a grade above 80

Preview of Questions Ends Here

Answers

Worksheet A1.1 Conversions of number systems

1. Complete the table below showing the value of each digit in the number 26951

Denary number	2	6	9	5
Power of 10	10^5	10^4	10^3	10^2
Value of power	100000	10000	1000	100
Calculation of digit's value	Two lots of 100000	Six lots of 10000	Nine lots of 1000	Five lots of 100
Total value of digit	200000	60000	9000	500

Each correct row gains a mark. 4 marks total.

2. Complete the table below showing the value of each digit in the number 11010

Binary number	1	1	0	1	
Power of 2	2^6	2^5	2^2	2^3	
Value of power	64	32	16	8	
Calculation of digit's value	One lot of 64	One lot of 32	No lots of 16	One lot of 8	No
Total value of digit	64	32	0	8	

Each correct row gains a mark. 4 marks total.

3. Convert the following numbers.
   a) 13 → **1101** [1]
   b) 45 → **101101** [1]
   c) 100 → **1100100** [1]
   d) 255 → **11111111** [1]
   e) 73 → **1001** [1]
   f) 1010 → **10** [1]
   g) 11001 → **25** [1]
   h) 11111111 → **2**
   i) 100000 → **32** [1]
   j) 101101 → **45** [1]

Each correct answer gains a mark. 10 marks total.

Worksheet A1.2 Use of binary to represent negative and floating-point numb

1. Convert the following 8-bit two's complement binary numbers into decimal.
   a) 11110011 = -13 [1]
   b) 10010101 = -107 [1]
   c) 11111111 = -1 [1]
   d) 11000000 = -64 [1]

2. Convert these denary negative numbers into 8-bit two's complement.
   a) -13 = 11110011 [1]
   b) -1 = 11111111 [1]
   c) -104 = 10011000 [1]
   d) -49 = 11001111 [1]

3. Convert the following 8-bit fixed-point binary numbers into denary.
   a) 0011.0000 = 3.0 [1]
   b) 1110.1000 = 14.5 [1]
   c) 0101.1000 = 5.5 [1]
   d) 1000.1000 = 8.5 [1]

4. Below are 8-bit floating-point numbers with the following format: 1 bit for sign, 4 bits for mantissa. Interpret their decimal value.
   a) 0 100 1010 = 26 [1]
   b) 1 011 0100 = -0.25 [1]
   c) 0 110 0000 = 0.25 [1]
   d) 1 010 1110 = -7.5 [1]

5. Convert the following denary numbers into 8-bit floating point using the forma
exponent, and 4 bits for mantissa.
   a) 3.5 = 0 100 1100 [1]
   b) -1.25 = 1 011 0100 [1]
   c) 0.5 = 0 010 0000 [1]
   d) 7.5 = 0 101 1110 [1]

Worksheet A1.3 Performing addition, subtraction, division and multiplicatior

Perform the following binary additions and convert your answers to denary.
1. 00101101 + 00010111 = 01000100 → Denary = 68 [2]
2. 01100011 + 00011100 = 10000111 → Denary = 127 [2]
3. 00011110 + 00010010 = 00111100 → Denary = 60 [2]

Perform the following binary subtractions and convert your answers to denary.
4. 01011010 − 00010101 = 01000101 → Denary = 69 [2]
5. 00111000 − 00011000 = 00100000 → Denary = 32 [2]
6. 01000010 − 00011110 = 00100100 → Denary = 36 [2]

Perform the following binary multiplications and convert your answers to denary.
7. 00000101 × 00000011 = 00001111 → Denary = 15 [2]
8. 00000110 × 00000010 = 00001100 → Denary = 12 [2]
9. 00001001 × 00000011 = 00011011 → Denary = 27 [2]

Perform the following binary divisions and convert your answers to denary.
10. 00110000 ÷ 00000010 = 00011000 → Denary = 24 [2]
11. 00010100 ÷ 00000010 = 00001010 → Denary = 10 [2]
12. 00100110 ÷ 00000010 = 00010011 → Denary = 19 [2]
13. What would it mean when performing subtraction, and calculation produces a r
two's complement?
It means the result is negative. [1]

Worksheet A1.4 Binary shifts

Shift the binary numbers to the left as indicated. Then convert to denary.
1. 00001101 << 1 = 00011010 → Denary = 26 [1]
2. 00001101 << 2 = 00110100 → Denary = 52 [1]
3. 00010010 << 3 = 10010000 → Denary = 144 [1]
4. 00100101 << 1 = 01001010 → Denary = 74 [1]

Shift the binary numbers to the right as indicated. Then convert to denary.
5. 01110000 >> 1 = 00111000 → Denary = 56 [1]
6. 00011000 >> 2 = 00000110 → Denary = 6 [1]
7. 10100000 >> 3 = 00010100 → Denary = 20 [1]
8. 00001111 >> 2 = 00000011 → Denary = 3 [1]

9. What is the effect of a 1-bit left shift on a binary number?
It multiplies the number by 2. [1] Each bit moves one place to the left, and a 0 is
Max 1 mark.

10. What is the effect of a 2-bit right shift on a binary number?
It divides the number by 4 (2^2), rounding down if needed. [1] Each bit moves two
in from the left. [1] *Max 1 mark.*

11. Explain why left shifts are used to multiply a number by powers of 2.
Each left shift moves bits one place higher in value. [1] So shifting left by 1 =
Max 1 mark.

12. a) The 8-bit binary number 00110100 is shifted left by 2 bits. What happens t
They are lost and cannot be recovered. [1] Data overflow can occur. [1]
Max 1 mark.
   b) What should you do with the empty spaces on the right?
Fill them with 0s. This keeps the number 8 bits long. [1]

Preview of Answers Ends Here

Worksheet A1.1 Conversions of number systems

Conversion of numbers between number systems.

1. Complete the table below showing the value of each digit in the number 2

Denary number	1 2	6	9	5
Power		10^4		
Value of power	100000		1000	
Calculation of digit's value		Six lots of 10000		
Total value of digit		60000		

2. Complete the table below showing the value of each digit in the number 1

Binary number	1	1	0	1	
Power of 2		2^5		2^3	
Value of power	64				
Calculation of digit's value			No lots of 16		
Total value of digit	64				

3. Convert the following numbers.

a) Convert **13** to binary .....

b) Convert **45** to binary .....

c) Convert **100** to binary .....

d) Convert **255** to binary .....

e) Convert **73** to binary .....

f) Convert **1010** to de

g) Convert **11001** to d

h) Convert **11111111** t

i) Convert **100000** to

j) Convert **101101** to

Worksheet A1.2 Use of binary to represent negative and f

Use of binary to represent negative and floating-point numbers.

1. Convert the following 8-bit two's complement binary numbers into decim
   a) 11110011 = _____
   b) 10010101 = _____
   c) 11111111 = _____
   d) 11000000 = _____

2. Convert these denary negative numbers into 8-bit two's complement.
   a) -13 = _____
   b) -1 = _____
   c) -104 = _____
   d) -49 = _____

3. Convert the following 8-bit fixed-point binary numbers into denary.
   a) 0011.0000 = _____
   b) 1110.1000 = _____
   c) 0101.1000 = _____
   d) 1001.1100 = _____

4. Below are 8-bit floating-point numbers with the following format: 1 bit fo
   4 bits for mantissa. Convert their decimal value.
   a) 0 1 0 = _____
   b) 1 011 0100 = _____
   c) 0 110 0000 = _____
   d) 1 010 1110 = _____

5. Convert the following denary numbers into 8-bit floating point using the f
   exponent, and 4 bits for mantissa.
   a) 3.5 = _____
   b) -1.25 = _____
   c) 0.5 = _____
   d) 7.5 = _____

Worksheet A1.3 Performing addition, subtraction, divisio

Binary arithmetic (addition, subtraction, multiplication and divisi

Perform the following binary additions and convert your answers to denary.

1. 00101101 + 00010111 = _____ → Den_____ = _____
2. 01100011 + 00011100 = _____ → Denary = _____
3. 00011110 + 000111[illegible] = _____ → Denary = _____

Perform the [illegible]ng binary subtractions and convert your answers to denary.

4. 01011010 − 00010101 = _____ → Denary = _____
5. 00111000 − 00011000 = _____ → Denary = _____
6. 01000010 − 00011110 = _____ → Denary = _____

Perform the following binary multiplications and convert your answers to dena

7. 00000101 × 00000011 = _____ → Denary = _____
8. 00000110 × 00000010 = _____ → Denary = _____
9. 00001001 × 00000011 = _____ → Denary = _____

Perform the following binary divisions and convert your answers to denary.

10. 00110000 ÷ 00000010 = _____ → Denary = _____
11. 000101[illegible] = _____ → Denary = _____
12. 00100110 ÷ 00000011 = _____ → Denary = _____

13. What would it mean when performing subtraction, and calculation produc leading 1 in two's complement?

.....

.....

Worksheet A1.4 Binary shifts

Shift the binary numbers to the left as indicated. Then convert to denary.

1. 00001101 << 1 = _____ → Denary = _____
2. 00001101 << 2 = _____ → Denary = _____
3. 00010010 << 3 = _____ → Denary = _____
4. 00100101 << 1 = _____ → Denary = _____

Shift the binary numbers to the right as indicated. Then convert to denary.

5. 01110001 >> 1 = _____ → Denary = _____
6. 00011000 >> 2 = _____ → Denary = _____
7. 10100000 >> 3 = _____ → Denary = _____
8. 00001111 >> 2 = _____ → Denary = _____

9. What is the effect of a 1-bit left shift on a binary number?

.....

.....

10. What is the effect of a 2-bit right shift on a binary number?

.....

.....

11. Explain how left shifting can be used to multiply a number by powers of 2.

.....

.....

12. The 8-bit binary number 00110100 is shifted left by 2 bits.
    a) What happens to the bits that "fall off" the left?

    .....

    .....

    b) What should you do with the empty spaces on the right?

    .....

    .....

Worksheet A1.5 Overflow errors

1. What is an overflow error?

   .....

   .....

   .....

2. Do these binary additions cause an overflow in 8 bits?

   a) 11111111 + 00000001 → YES / NO

   b) 10000000 + 10000000 → YES / NO

   c) 00001111 + 00000001 → YES / NO

   d) 01111111 + 00000001 → YES / NO

3. List two problems an overflow error could cause in real life.

   1. .....

   2. .....

4. What are two ways to prevent or handle overflow?

   1. .....

   2. .....

5. Imagine a video game where the score is stored as 8-bit.
   What happens if the score goes from 255 to 256?

   .....

   .....

Worksheets A2.1 to A2.3 Operators

Relational operators compare values and return True or False.

1. Fill in the blanks with the correct relational operator so that the output is

   a) 7 ___ 7 → True

   b) 4 ___ 10 → True

   c) 9 ___ 3 → True

   d) 6 ___ 6 → True

2. What will each of the following expressions return?

   a) 5 == 5 → _____

   b) 10 < 3 → _____

   c) 8 >= 8 → _____

   d) 4 != 2 → _____

Boolean operators combine multiple conditions.

3. Fill in the blanks with the correct Boolean operator

   a) True ___ False → _____

   b) False ___ False → _____

   c) ___ True → _____

   d) (5 > 2) ___ (2 == 2) → _____

4. Evaluate the following and write the output.

   a) (6 > 2) and (3 < 5) → _____

   b) not (4 == 4) → _____

   c) (7 < 2) or (8 == 8) → _____

5. Fill in the blanks with the correct arithmetic operator.

   a) 10 ___ 3 gives → 1

   b) 7 ___ 2 gives → 3

   c) 3 ___ 2 equals 2 → 1.5

   d) 3 ___ 5 equals 15 → 15

6. Write the result of the following expressions.

   a) 8 + 4 * 2 = _____

   b) 9 % 4 = _____

   c) 15 // 4 = _____

   d) (6 – 2) * 3 = _____

Worksheets A2.4 and A2.5 Variables

Data types used in programming (string, integer, float/real, Boole
Declaration, initialisation, and use of constants and variables.

1. Match each data type to the correct description.

	Data type
a)	String
b)	Int
c)	Fl
d)	Boolean

	Description
1)	A number with a decimal point (e
2)	A value that can only be True or
3)	A group of characters or text (e.g
4)	A whole number (e.g. 7, -3)

2. Circle the correct data type for each value.

   a) "123" → String / Integer / Boolean

   b) True → String / Boolean / Float

   c) 7.0 → Float / Integer / String

   d) "False" → Boolean / String / Float

3. Fill in the blanks with the correct data type.

   a) "Hello world" is a _____

   b) 25 is a ex _____ an _____.

   c) Fals _____ value.

   d) 3.1415 is a _____.

4. Fill in the blanks.

   a) A variable is a named space in memory that can _____.

   b) A constant is a named value that _____ change during a program.

   c) The process of giving a variable a value when it is created is called _

   d) Declaring a variable means _____.

5. Write down what each line of code is doi g.

   a) name = "Alice" → _____

   b) Pi → _____

   c) scor 0 → _____

   d) is_active = True → _____

Worksheet A3.5 Local and global variables

User defined functions. Local and global variables.

Using the program ***student_reports.py***, complete the following tasks.

1. Identify all user-defined **function names**, including the line number they a

2. State the purpose of each user-defined **function**.

3. Identify all **parameters** in the user-defined functions, including the line nu

4. Identify all **arguments** in the user-defined functions, including the line nu

5. Identify all **global variables**, including the line number they appear on.

6. Identify all **local variables**, including the line number they first appear on.

Worksheet B1.3 Stacks and queues

1. A stack is a data structure that follows the _____ rule.
2. A queue is a data structure that follows the _____ rule.
3. In a stack, the most recently added item is the _____ one to be rem
4. In a queue, the item added first is the one r_____ve_____.
5. Which data structure would be more useful in the scenarios below?

Scenario	Stack or queue
Printer job list	
Undo feature in a text editor	
Customers lining up at a checkout	
Website back-button history	

6. Describe **one reason** why a programmer would use a stack instead of a qu

.....

.....

.....

.....

7. Given t_____owing actions, state the final contents of the stack and queue
   - Add A
   - Add B
   - Add C
   - Remove one item

   Stack contents: .....

   Queue contents: .....

Worksheets B2.1 and B2.2: Input and output; Numerical

1. To use the random module, you must first write:

   _____ `random`

2. `random.randint(1, 5)` could r... which of the following?
   a) ☐ 1 b) ☐ 7.5 c) ☐ 6 d) ☐ 5

3. What do... `...hoice(['red', 'blue', 'green'])` do?

   .....

   .....

   .....

4. What does `random.shuffle(my_list)` do?

   .....

   .....

   .....

5. Write code to do the following.
   a) Print a random number between 10 and 20

      .....

      .....

      .....

      .....

   b) Shuffle the list ['apple', 'banana', 'cherry'] and print the result

      .....

      .....

      .....

      .....

      .....

      .....

   c) Pick a random letter from the w... 'lo...g'

      .....

      .....

      .....

      .....

      .....

      .....

6. Match the function to its output.

a)	round(7.6)
b)	int(7.6)
c)	round(3.5)
d)	round(2.5)

1)	7
2)	2
3)	8
4)	4

7. Predict the output.

   a) int(→ _____

   b) round(4.5) → _____

   c) math.floor(-3.2) → _____

   d) math.ceil(-3.2) → _____

8. What is the difference between truncating and rounding?

.....

.....

.....

.....

.....

.....

Worksheet B2.3: String handling

1. What will the following code output?

```
first = "Good"
second = "Morning"
message = first + " " + second
print(message)
```

Answer: _____

2. Fill in the blank to join the three words.

```
word1 = "Computer"
word2 = "Science"
word3 = "Rocks"
sentence = _____
print(sentence)
```

3. What does .split() do?

	Tick (✓)
Joins strings together	
Removes punctuation	
Breaks a string into a list of words	

4. What will the code output?

```
sentence = "I love Python"
words = sentence.split()
print(words)
```

Answer: _____

5. Turn this string into a list using a built-in function.

```
shopping = "milk,bread,eggs"
```

.....

.....

.....

6. Complete the code to clean up the user's name.

```
name = input("Enter your name: ")
clean_name = name._____().._____()
print(f"Hello, {clean_name}!"
```

7. Match the method to its use.

Method		Use	
a)	`.strip()`	1)	Capitalises first letter of each wo
b)	`.lower()`	2)	Makes everything lowercase
c)	`.title()`	3)	Rem v s ra spaces from input

8. Fill in the missing line ... name and age clearly using a formatted str

```
name = "Zara"
age = 17
print(_____________________________)
# Expected: Zara is 17 years old.
```

9. True or False?

`f"{name} is {age} years old"` is an example of a formatted strin

True ☐ False ☐

10. Write a program that:
    - Takes a full sentence as input
    - Splits it into words
    - Counts, and outputs, how many wo s
    - Prints each word on a ne

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Worksheet B2.5: External text files

1. Define the term delimiter in the context of text files.

.....

.....

.....

.....

2. Underline the delimiter in each example below.
   a) Alice|Maths
   b) John,88,Physics
   c) 101|Bobby|History

3. Draw each of the following delimiters.
   a) Comma
   b) Pipe

4. Write one line of code to open a file called data.txt in write mode.

.....

5. Complete this code to write three lines into a file called greetings.txt

```
with open("greetings.txt", "w") as f:
    f.write("Hello\n")
    _____
    _____
```

6. Given this code, what will be printed?

```
with open("scores.txt", "r") as f:
    for line in f:
        name, score = line.strip().split(",")
        print(name)
```

scores.txt contents:

```
Asha,15
Brian,14
```

.....

.....

7. List two benefits of using text files to store program data.

.....

.....

.....

.....

8. List two drawbacks of using .txt files.

9. Would a .txt file be a good choice for storing users' login details in a web a

10. Write a program that reads a file called students.txt which has:
    - each line is in the format name,grade
    - prints only the names of students who got a grade above 80

	Algorithm	P

12	array ← [64, 34, 25, 12, 22, 11, 90]
	n ← LEN(array)
	FOR i ← 0 TO n - 2
	  FOR j ← 0 TO n - i - 2 DO
	  IF array[j] > array[j + 1] THEN
	  temp ← array[j]
	  array[j] ← array[j + 1]
	  array[j + 1] ← temp
	  ENDIF
	  NEXT j
	NEXT i
	OUTPUT "Sorted array: " & array	

13	PROCEDURE greet()
	  OUTPUT "Welcome to the program!"
	END PROCEDURE
	
	FUNCTION add(num1, num2)
	  RETURN num1 + num2
	END FUNCTION
	
	greet()
	OUTPUT "Enter first number:"
	a ← INPUT
	OUTPUT "Enter second number:"
	b ← INPUT
	result ← add(a, b)
	OUTPUT "The sum is: " result	

Preview of Questions Ends Here