

Computational Maths

**CT4032**

**Workbook – Week 2**

2023 – 24

**Dr Nasreen Anjum**

School of Computing and Engineering

**University of Gloucestershire 2023/24**

All rights reserved. No part of this publication may be reproduced, stored or transmitted in any form or by any means, including – but not limited to – photocopy, recording, or any information storage and retrieval system, without the specific prior written permission of University of Gloucestershire.

1

# Coding Challenge-1

**Activity: Even or Odd Number Checker**

**Objective:** Create a Python program that checks if a given number is even or odd.

**Instructions:**

1. Display a welcome message explaining the purpose of the program.
2. Ask the user to input a whole number (an integer).
3. Check if the input number is even or odd. An even number is divisible by 2 without a remainder, while an odd number is not.
4. Display the result, indicating whether the number is even or odd.
5. Ask the user if they want to check another number or exit the program.
6. If the user chooses to check another number, loop back to step 2. If they choose to exit, display a thank you message and end the program.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Coding Challenge-2

**Activity: Perfect Square Checker**

**Objective:** Create a Python program that checks if a given number is a perfect square or not.

**Instructions:**

1. Display a welcome message explaining the purpose of the program.
2. Ask the user to input a whole number (an integer).
3. Check if the input number is a perfect square. A perfect square is a number that can be expressed as the square of an integer.
4. Display the result, indicating whether the number is a perfect square or not.
5. If the user chooses to check another number, loop back to step 2. If they choose to exit, display a thank you message and end the program.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Coding Challenge-3

**Activity: Cube Number Checker**

**Objective:** Create a Python program that checks if a given number is a cube number or not.

**Instructions:**

1. Display a welcome message explaining the purpose of the program.
2. Ask the user to input a whole number (an integer).
3. Check if the input number is a cube number. A cube number is a number that can be expressed as the cube of an integer.
4. Display the result, indicating whether the number is a cube number or not.
5. If the user chooses to check another number, loop back to step 2. If they choose to exit, display a thank you message and end the program.

# Coding Challenge-4

**Activity: Triangular Number Display**

**Objective:** Create a Python program that displays triangular numbers up to a given number.

**Instructions:**

1. Display a welcome message explaining the purpose of the program.
2. Ask the user to input a positive whole number (an integer).
3. Check if the input number is positive. If it's not, ask the user to re-enter the number until a positive number is provided.
4. Generate and display triangular numbers up to the given number. A triangular number is a number that can be represented as the sum of natural numbers from 1 to n.
5. For each triangular number, also display the value of 'n' that corresponds to it.
6. Ask the user if they want to display triangular numbers up to another number or exit the program.
7. If the user chooses to display triangular numbers for another number, loop back to step 2. If they choose to exit, display a thank you message and end the program.

# Coding Challenge-5

**Activity: Perfect Number Checker**

**Objective:** Create a Python program that checks if a given number is a perfect number or not.

**Instructions:**

1. Display a welcome message explaining the purpose of the program.
2. Ask the user to input a positive whole number (an integer).
3. Check if the input number is positive. If it's not, ask the user to re-enter the number until a positive number is provided.
4. Determine if the input number is a perfect number. A perfect number is a positive integer that is equal to the sum of its proper divisors (excluding itself).
5. Display the result, indicating whether the number is a perfect number or not.
6. If the user chooses to check another number, loop back to step 2. If they choose to exit, display a thank you message and end the program.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Coding Challenge-5

Coding Challenge: Prime Number Generator

Write a Python program to generate a list of prime numbers within a specified range. Prime numbers are positive integers greater than 1 that have exactly two distinct positive divisors: 1 and themselves. Your program should efficiently identify and list all prime numbers in the given range.

**Congratulations on successfully completing all of the coding challenges!**

*Your dedication and effort in solving these challenges demonstrate your commitment to learning and improving your programming skills. It's a significant achievement to tackle a variety of coding problems and develop practical solutions using Python.*

*By working through these activities, you've gained valuable experience in areas like problem-solving, algorithmic thinking, user input handling, and conditional statements. These skills are essential for any programmer and will serve as a strong foundation for your future coding endeavors.*

*Keep up the fantastic work and continue to challenge yourself with more coding tasks. Each challenge you conquer brings you one step closer to becoming a proficient coder. Remember that learning is a continuous journey, and the skills you've acquired will undoubtedly benefit you in your academic and professional pursuits.*

*Stay motivated, keep coding, and reach for even greater heights in your programming journey!*