**National University of Computer & Emerging Sciences (NUCES) Islamabad,**

Department of Computer Science

**Programming Fundamentals – Fall 2022**

**LAB 11**

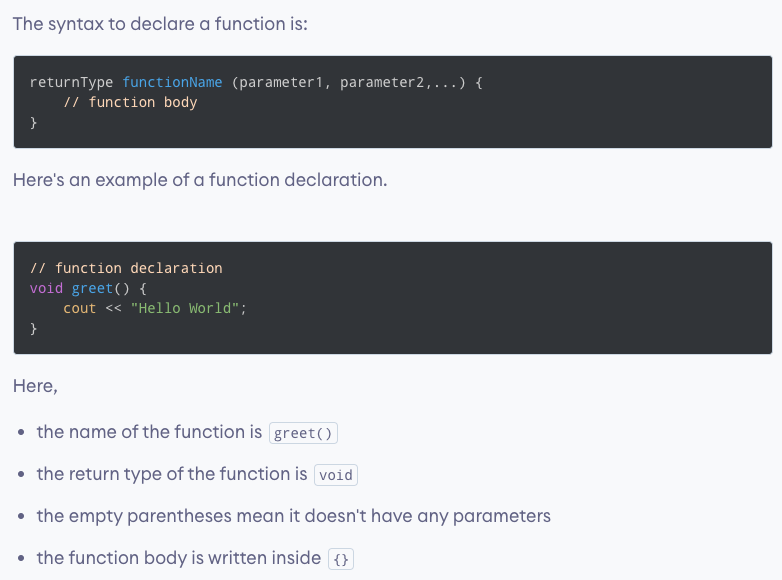


**Learning Outcomes**

In this lab you are expected to learn the following:

* Functions (definition, calling, forward declaration, default argument, overloading)

**Syntax and Basic Example of a Function:**



**Graphical user interface, text, application

Description automatically generated**

**How does the above program work?**

**Diagram

Description automatically generated**

**Function example with parameters:**

**Text

Description automatically generatedGraphical user interface, text, application

Description automatically generated**

**How does the above program work?**

**Diagram

Description automatically generated**

**Function example with return type:**

**Text

Description automatically generatedGraphical user interface, text

Description automatically generated with medium confidence**

**How does the above program work?**

**A picture containing timeline

Description automatically generated**

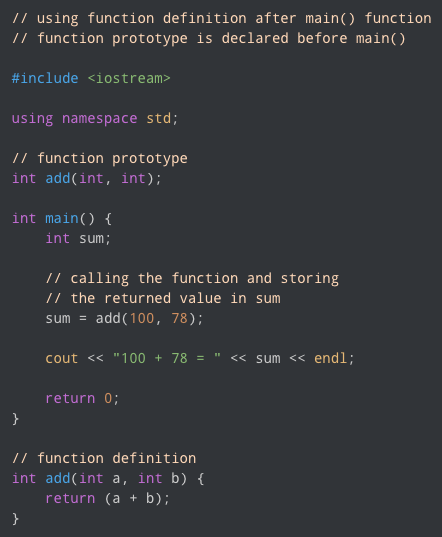
**Function Prototype:**

**Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated**

**Function Example with Prototype:**

****

**Function Overloading**

**Graphical user interface, application, Teams

Description automatically generated**

**Function Overloading using different types of parameters:**

**Text

Description automatically generated**

**Shape, rectangle

Description automatically generated**

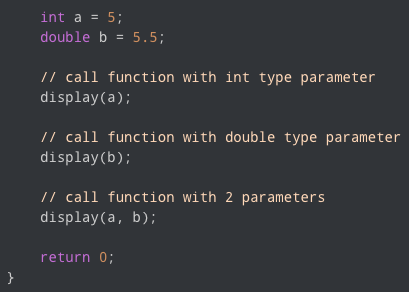
**How does the above program work?**

**A picture containing diagram

Description automatically generated**

**Function Overloading using different number of parameters:**

**Text

Description automatically generatedText

Description automatically generated**

**How does the above program work?**

**A picture containing timeline

Description automatically generated**

**Default Arguments**

**Graphical user interface, text, application, email

Description automatically generated**

**Working of default arguments:**

**Graphical user interface

Description automatically generated with medium confidence**

**Graphical user interface, application

Description automatically generated**

**Application

Description automatically generated with low confidence**

**Example of default argument:**

**Text

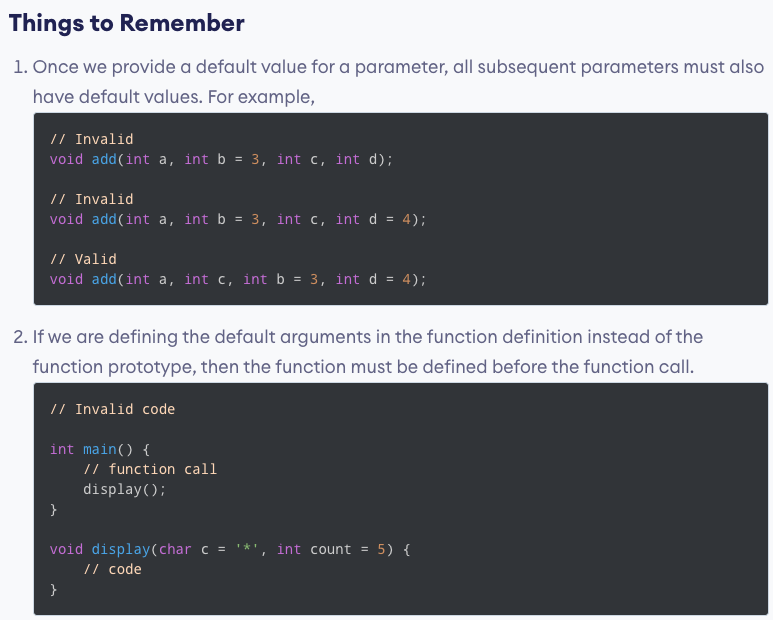
Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Text

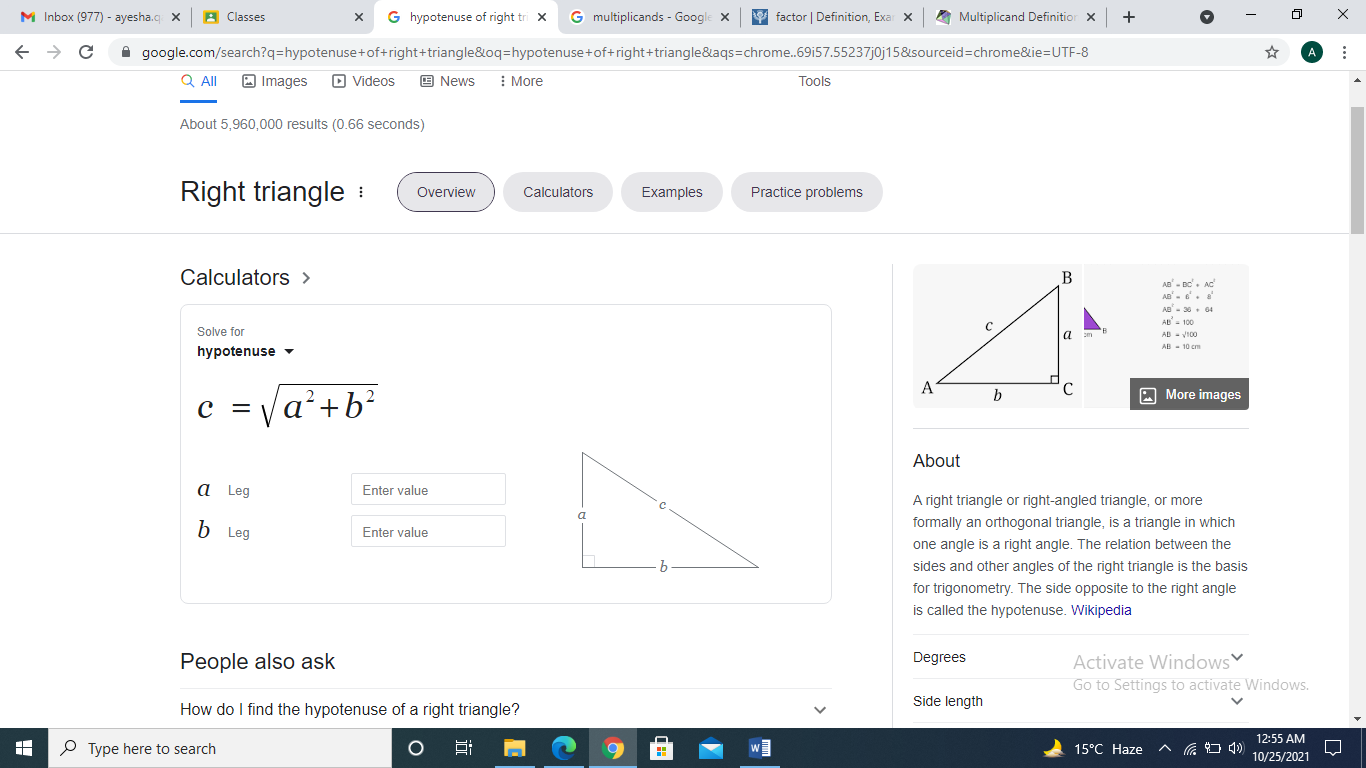
Description automatically generated**

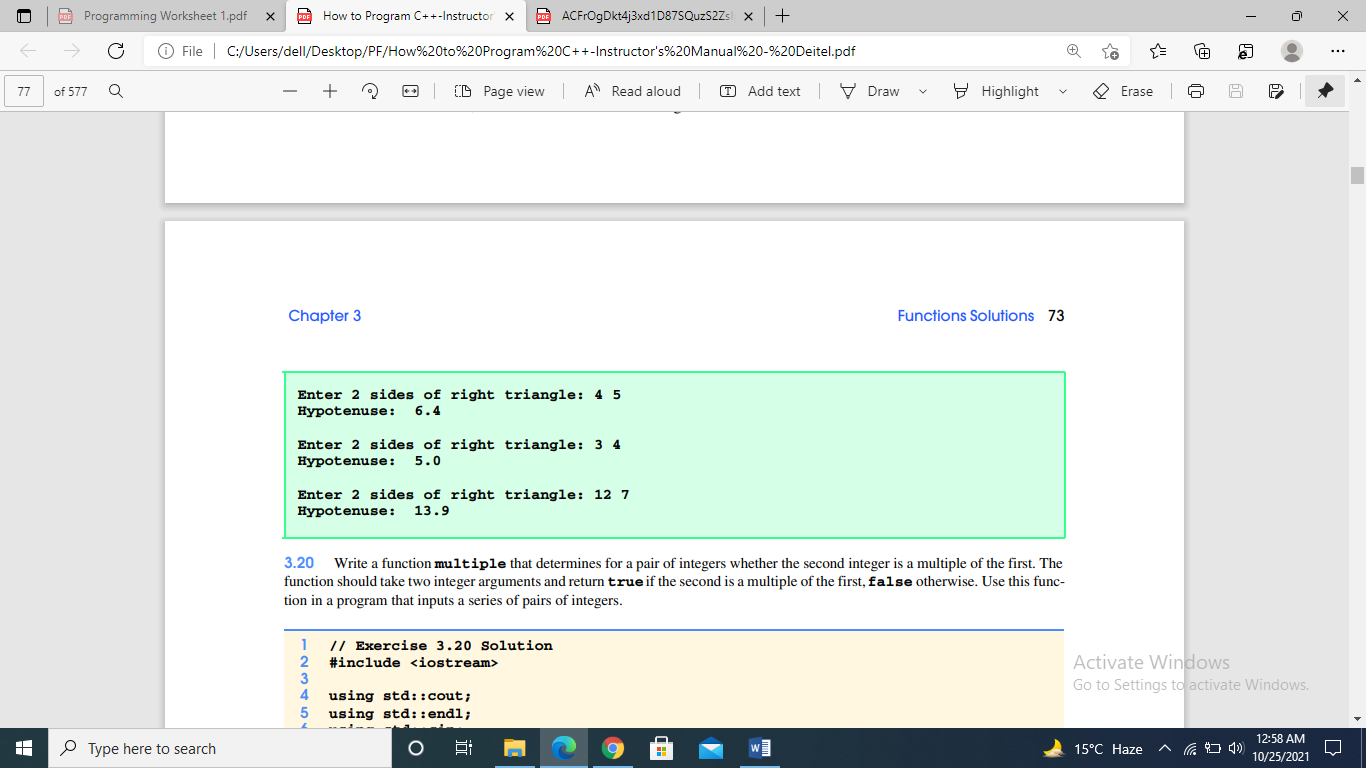
****

**Task 1:**

Define a function **hypotenuse** that calculates the length of the hypotenuse ‘c’ of a right triangle when the other two sides ‘a’ and ‘b’ are given. Use this function in a program using the loop to determine the length of the hypotenuse ‘c’ for each of the following triangles. The function should take two sides as arguments of type **double** and return the hypotenuse as a **double.** The prototype of the function is: **double hypotenuse ( double, double );**

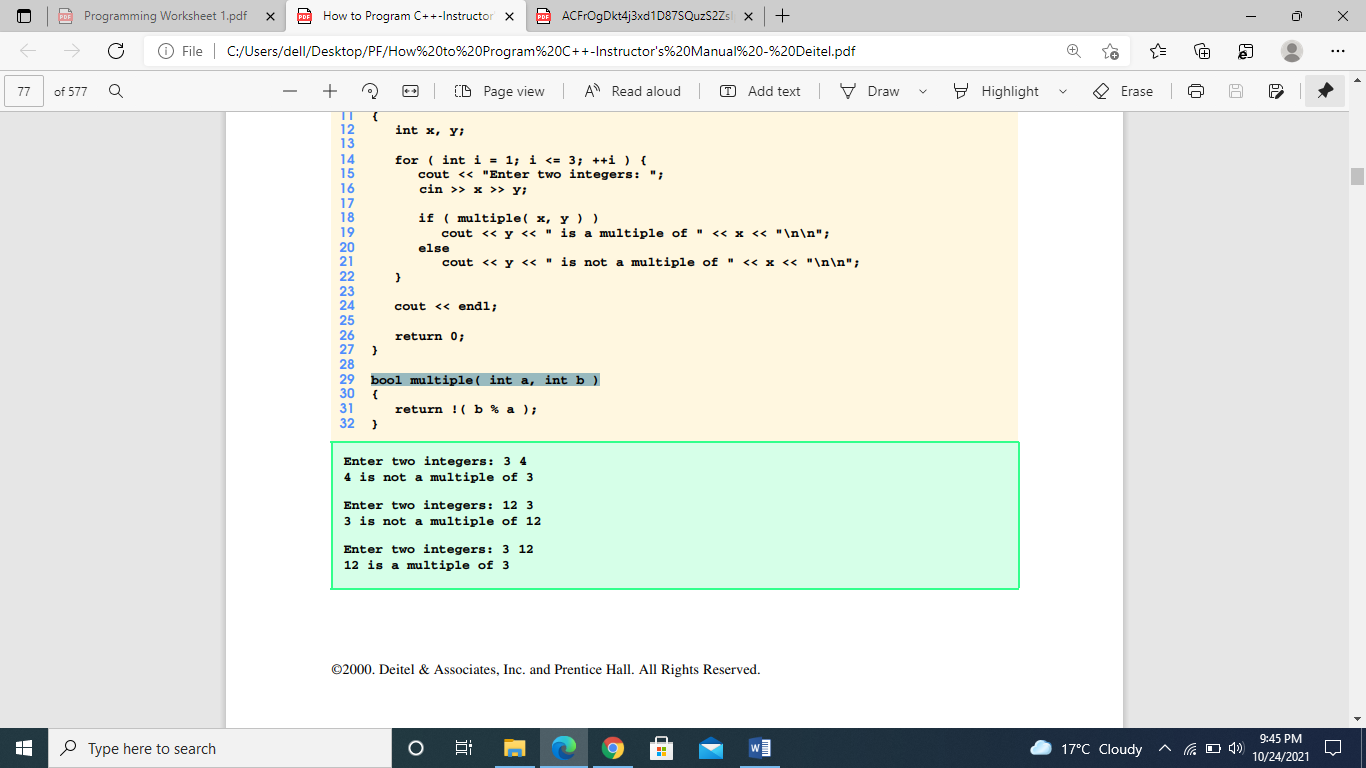
The formula for hypotenuse calculation is as follows:





**Task 2:**

Write a function **multiple** that determines for a pair of integers whether the second integer is a multiple of the first. The function should take two integer arguments and return **true** if the second is a multiple of the first, **false** otherwise. Use this function in a program that inputs a series of pairs of integers. The prototype of the function is: **bool multiple ( int, int );**



**Task 3:**

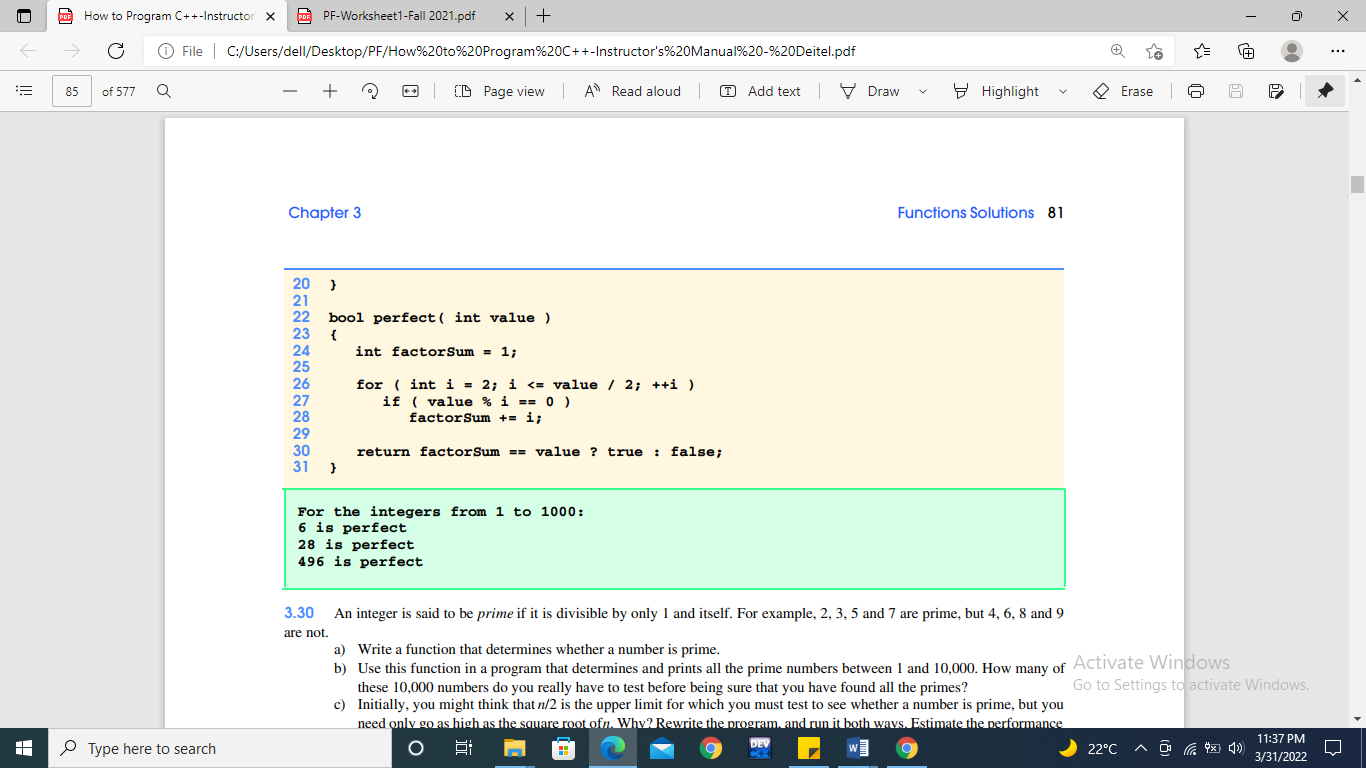
An integer number is said to be a perfect number if the sum of its factors, including 1 (but not

the number itself), is equal to the number. For example, 6 is a perfect number, because 6 = 1 + 2

+ 3. Write a function **perfect( )** that determines whether parameter number is a perfect number.

Use this function in a program that determines and prints all the perfect numbers between 1 and

1000. The prototype of the function is: **bool perfect ( int );**



**Task 4:**

Given the dimensions of the 3D Shapes like **Cube**, **Cuboid**, or **Cylinder**, the task is to find the volume of all the 3D Shapes using Function Overloading.

**Help:** The given problem can be solved by creating the different functions having the same name, say **Volume** but different function definitions. For each shape and then overload all the functions by passing different parameters to them and then call all the functions from the main function one after another.

**Input:**

Cube: L = 2,

Cuboid: L = 3, B = 2, H = 3,

Cylinder: R = 1, H = 7  
**Output:**  
Volume of Cube: 8  
Volume of Cuboid: 18  
Volume of Cylinder: 22