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

Department of Computer Science

**Programming Fundamentals – Fall 2022**

**LAB 12**



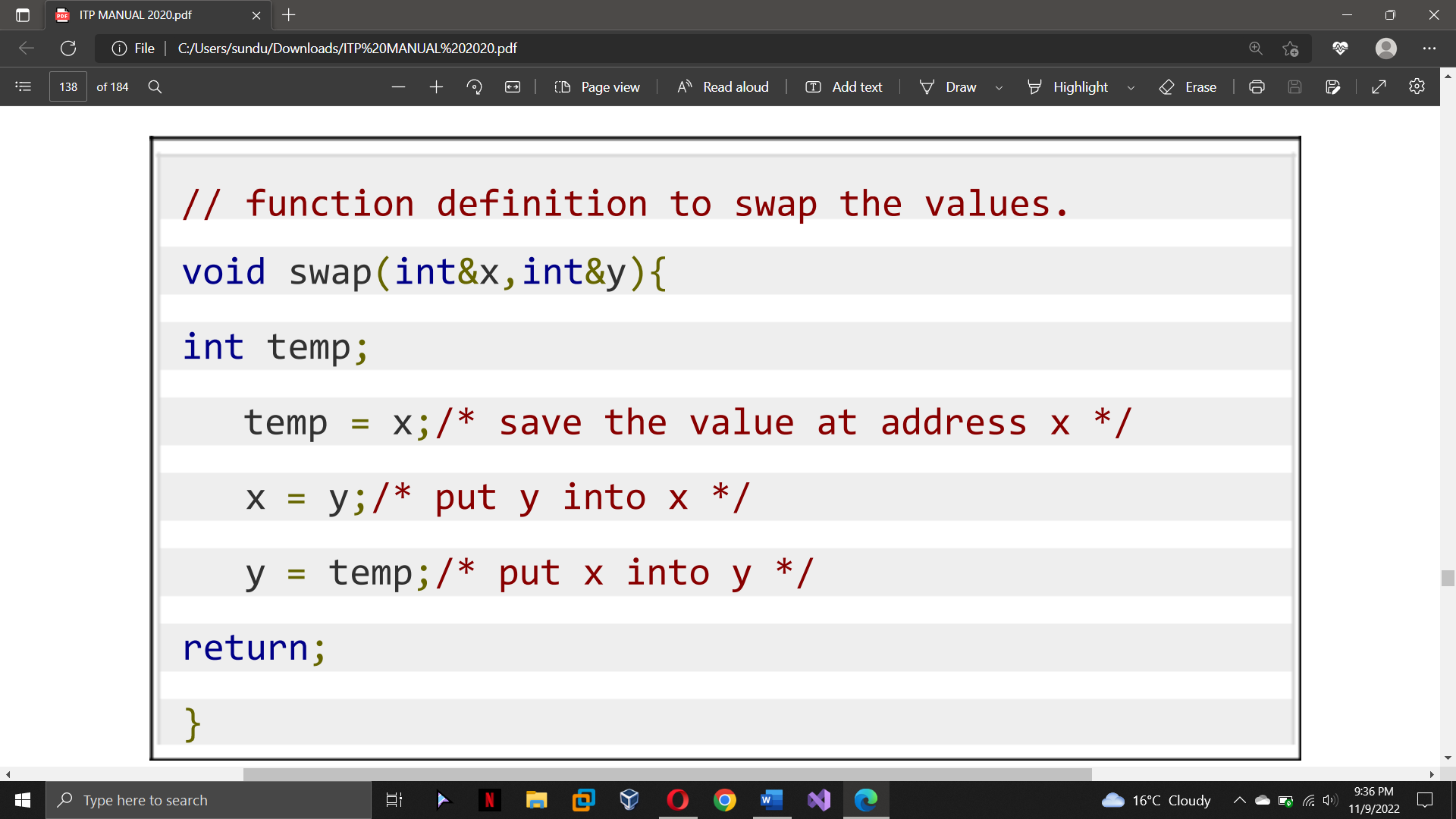
**Learning Outcomes**

In this lab you are expected to learn the following:

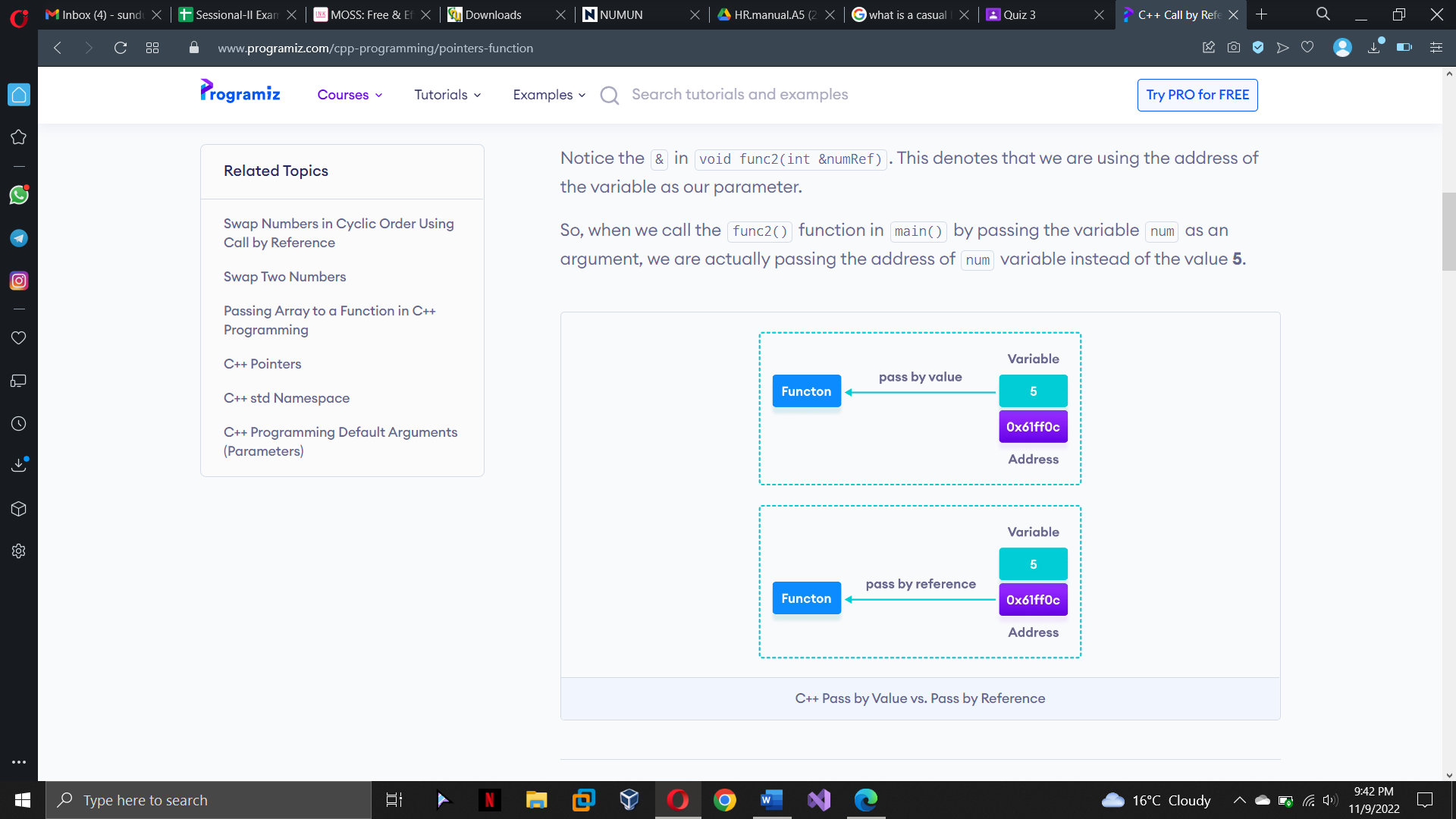
* Functions calling by reference
* Functions calling one function in the other.
* Static and global variables

**Functions calling by reference**

The call by reference method of passing arguments to a function copies the reference of an argument into the formal parameter. Inside the function, the reference is used to access the actual argument used in the call. This means that changes made to the parameter affect the passed argument. To pass the value by reference, argument reference is passed to the functions just like any other value. So accordingly you need to declare the function parameters as reference types as in the following function swap(), which exchanges the values of the two integer variables pointed to by its arguments.



Notice the ‘&’ in void swap(int &x, int &y). This denotes that we are using the address of the variable as our parameter. So, when we call the swap() function in main() by passing the variable num as an argument.



Example:

#include <iostream>

using namespace std;

// function definition to swap values

void swap(int &n1, int &n2) {

int temp;

temp = n1;

n1 = n2;

n2 = temp;

}

int main()

{

// initialize variables

int a = 1, b = 2;

cout << "Before swapping" << endl;

cout << "a = " << a << endl;

cout << "b = " << b << endl;

// call function to swap numbers

swap(a, b);

cout << "\nAfter swapping" << endl;

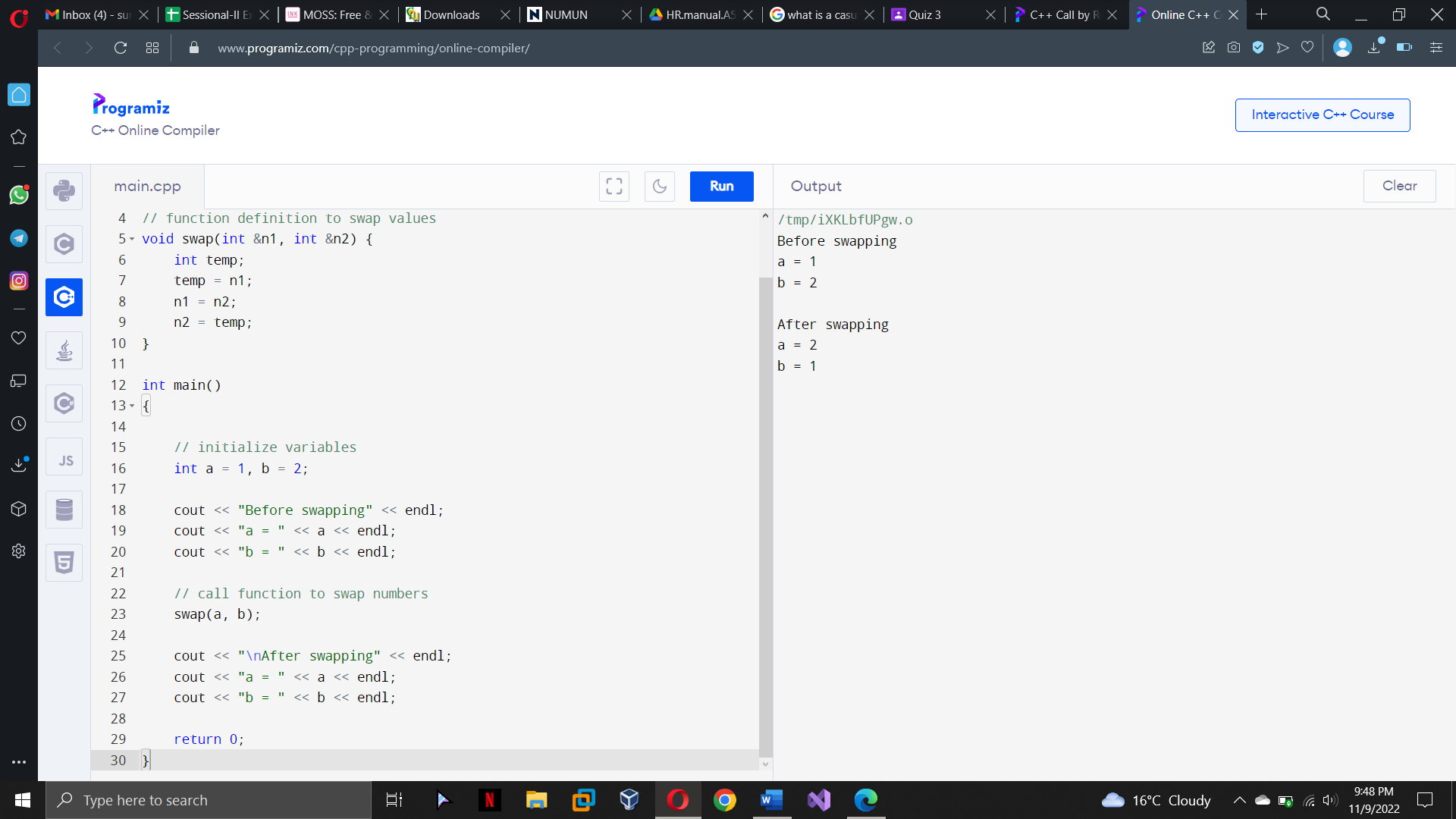
cout << "a = " << a << endl;

cout << "b = " << b << endl;

return 0;

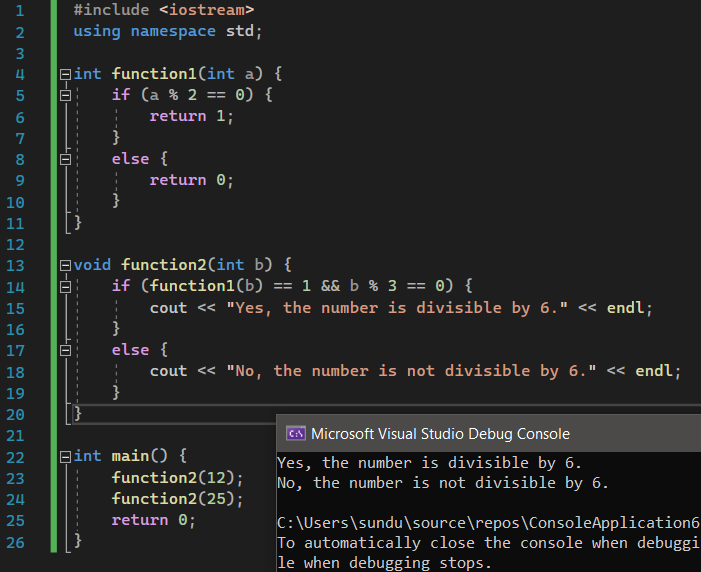
}

Output:



**Functions calling one function in the other:**

We can call a function inside another function. Notice, we were calling our functions inside the main function. Now look at an example in which there are two user defined functions. And we will call one inside another.



A number is divisible by 6, if it is divisible by both 2 and 3. We have a function function1 which will return 1 if the given number is divisible by 2. Another function that we have defined is fuction2 which calls function1 inside itself.

If ( function1(b)==1 && b%3 == 0 ) - In our case, b and thus a are 12 in the first case and 25 is the next case. So, if function1 returns 1, the number is divisible by 2. And if b%3==0 is true, 'b' is divisible by 3. So, if the number is divisible by both 2 and 3, then it is divisible by 6 also.

**Static and global variables**

**Static Variable**

Keyword static is used for specifying a static variable. For example:

int main()

{

static float a;

}

A static local variable exists only inside a function where it is declared (similar to a local variable) but its lifetime starts when the function is called and ends only when the program ends. The main difference between local variable and static variable is that, the value of static variable persists the end of the program.

Example:

#include <iostream>

using namespace std;

void test()

{

// var is a static variable

static int var = 0;

var = var+2;

cout << var << endl;

}

int main()

{ test(); }

Output: 2

**Global Variable**

If a variable is defined outside all functions, then it is called a global variable. The scope of a global variable is the whole program. This means, it can be used and changed at any part of the program after its declaration. Likewise, its life ends only when the program ends.

#include <iostream>

using namespace std;

// Global variable declaration

int c = 12;

void test();

int main()

{

c = c + 1 ;

// Outputs 13

cout << c <<endl;

test();

return 0;

}

void test()

{

c = c + 1 ;

// Outputs 14

cout << c;

}

Output :

13

14

**Task 1:**

Write a C++ program that takes two integers. By reference pass the two numbers to a function that will reduces them to the lowest possible form by dividing the two with each other.

Sample Input: a= 50 , b=5;

Sample Output: a=10, b=1

**Task 2:**

Write a C++ program that has a function with 3 parameters, one will be passed by value and the other two will be passed by reference. The user enters a 32-bit integer value and is passed to the function by value. The function slips the 32bit number into two. The first half (16bits) will be saved in the second parameter which is passed by reference, while the other half (16bits) will be saved in the third parameter which is passed by reference.

**Task 3:**

Write a C++ program that inputs a number. Pass the reference of the number to a function that will calculate the reverse of the number

Sample Input: 123

Sample Output: 321

**Task 4:**

Write a C++ program that declares an integer as a global variable, the program inputs a number and store it in the global variable. The program than pass it to a function that will check that the entered number is greater than zero but less than 1000. If the number lies between the range, the function passes the value to another function that will display that the number is even or odd.