Polytechnic University of Puerto Rico
Electrical and Computer Engineering & Computer Science Department
CECS 2223 – Computer Programming I Lab

Lab 4

Name: Coral S. Schmidt ID#: 148830

1. Copy the source code developed for Lab 4 and paste it as **text** below. (*15 points*)

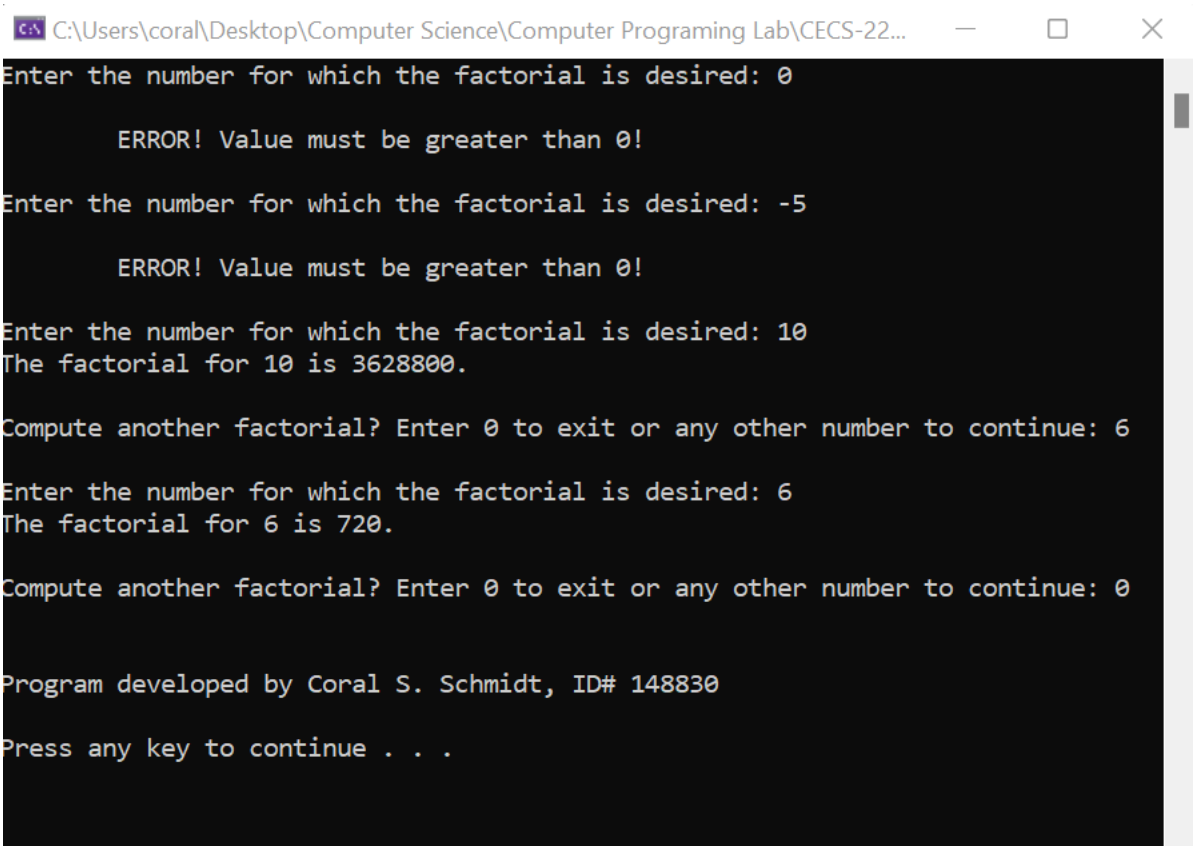
```
/*
* CECS 2203, Computer Programming I Laboratory
* Spring 2023, Sec. 06
* Date: April 13, 2023
* Topic: Lab 4 - Void and value returning methods
* File name: lab04.cpp
*
* Instructions and problem statement:
* In mathematics, the factorial is the product of all positive integers less than
and
* equal to the number. For example, the factorial for 3, denoted 3!, would be the
* product 1 x 2 x 3, resulting in 6. Your solution prompts the user for the number
which
* the factorial is desired, and prints the correct value.
*
* Complete the program by writing the correct C++ statements. Execute the program
and
* save a screenshot for your report.
*
* Name: Coral S. Schmidt Montilla, YOUR ID# 148830
*/

// write the appropriate preprocessor directive
#include <iostream>

// write the appropriate using directive
using namespace std;

// declare a method named getNumber which has no parameters and returns an integer
value
int getNumber();

// declare a method named factorial that receives the address of an integer and an
// integer value as parameters and has no return value
void factorial(int&,int);

// declare a method named printFactorial that has no parameters or retrun value
void printFactorial();

// declare a method named personalInfo which has no parameters or return value
void personalInfo();

int main() {
	// Write the statement that declares the integer variable seguir
	// and initializes it to 0
	int seguir = 0;
```

```
	// Develop a do/while iteration control structure which uses the value of
seguir
	// as the sentinel. First, call the printFactorial method, then ask the user
	// if another factorial is to be computed using the phrase
	// "Compute another factorial? Enter 0 to exit or any other number to
continue: ",
	// and save the value in seguir.
	// Test your code by calling printFactorial 3 times.
	do {
		// call the printFactorial method
		printFactorial();
		cout << endl;

		// ask the user if another factorial is to be computed
		cout << "Compute another factorial? Enter 0 to exit or any other number
to continue: ";

		// store the value in seguir
		cin >> seguir;
		cout << endl;

	} // The do/while cycle will continue if the value of seguir is not 0.
	while (seguir != 0);

	// call the personalInfo method
	personalInfo();

	system("pause"); // for Visual Studio only
	return 0;
}

// The getNumber has no parameters and returns an integer value. Declare a local
integer
// variable named valor and initialize it to 0. Prompt the user for a positive
integer,
// using the phrase "Enter the number for which the factorial is desired: " and
store
// the value in valor. Make sure that valor is greater than 0 using a while
iteration
// control structure. If the condition is not met, print the phrase
// "\n\tERROR! Value must be a greater than 0!\n\n". The function returns an integer
// greater than 0.
int getNumber()
{
	int valor = 0;

	cout << "Enter the number for which the factorial is desired: ";
	cin >> valor;

	while (valor <= 0) {

		cout << "\n\tERROR! Value must be greater than 0!\n\n";
		cout << "Enter the number for which the factorial is desired: ";
```

```
		cin >> valor;

	}

	return valor;
}


// The printFactorial method has no parameters or retrun value. Declare a local
integer
// variable named number and initialize it to the return value of the getNumber
method.
// Declare another local integer variable named facto, and initialize it to 0. Then
call
// the factorial method using number and facto as parameters. After returning from
the method,
// print the phrase "The factorial for [number] is [factorial].\n", susbtituting
[number]
// and [factorial] for the appropriate values.
void printFactorial() {

	int number = getNumber();
	int facto = 0;

	factorial(facto, number);

	cout << "The factorial for " << number << " is " << facto << ".\n";
}

// The factorial function computes the factorial for a number. It receives the
address of
// an integer, labeled as result, and an integer value, labeled as number, as
parameters
// and has no return value. Implement a for control structure to compute the
factorial,
// saving the intermediate product in result. The counter of the for cycle should be
used
// in the multiplication, so make sure to initialize it accordingly. The only
variable
// declared in this method is the counter of the for iteration control structure.
void factorial(int& result, int number) {
	result = 1;

	for (int i = 1; i <= number; ++i) {
		result *= i;
	}
}


// The personalInfo method has no parameters or return value. It prints the phrase:
// "Program developed by [YOUR NAME], ID#[YOUR ID NUMBER]", where the square
brackets
// and the text within is substituted with your personal information. Make sure to
add
```

```
// an empty line before and after the phrase.
void personalInfo()
{
	cout << endl;
	cout << "Program developed by Coral S. Schmidt, ID# 148830" << endl;
	cout << endl;
}
```

2. Paste the screenshots of the program's execution below. (*5 points*)



3. Comment on any warnings or errors revealed by Visual Studio. If any error messages were present, list the error and describe how you corrected it. If no errors or warnings were revealed, comment on the most important aspect of developing the solution. (*5 points*)

   The correct implementation of the methods is the most important aspect of developing the solution, as they are required for the program to run correctly. The getNumber method validates the input, the factorial method computes the factorial of a number, the printFactorial method combines the previous two methods to print the factorial of a number, and the personalInfo method prints the programmer's personal information. To

calculate and print factorials, the program also employs a do-while loop with a sentinel variable.