**LAB EXERCISE 3**

**TOPIC: FUNCTIONS**

**NAME: MAXIVIANNA BINTI ROBERT  
MATRIC NO: A24CS0109**

**SECTION: 02**

**QUESTION 1**

Describe the difference between predefined function and programmer-defined function?

* Predefined function is a built-in function in the compiler where the source code is already pre-determined and does not appear in the program while programmer-defined function is a function where all the statements are created by the programmer.

**QUESTION 2**

Write a statement to calculate the equation or to convert the statement below using function from library.

1. Square root of y.

return sqrt(x);

1. x to the power of y.

return pow(x,y);

1. cos x.

return cos(x);

1. Change character to uppercase.

char A = toupper(‘a’);

1. Copy the string of x into string y.

return strycpy(y,x);

**QUESTION 3**

What is the difference between local variable, global variable, global constant and static local variable?

* Local variable is defined inside a function while global variable is defined outside all functions. Local variables can only be accessed in the function while global variables can be accessed throughout the program.
* Global constants are global variables with fixed values that can be used throughout the program execution. Static local variables are local variables that retain their values between function calls.

**QUESTION 4**

Given the following coding, fill in the blank with the “terms” of function as a comment.

#include <iostream>

using namespace std;

int average(int, int, int); //Function prototype

int main()

{

int x, y, z, avrg;

cout << "Please enter three numbers:" << endl;

cin >> x >> y >> z;

avrg = average (x, y, z); //Function call

cout << "The average of the given three numbers is: " << avrg << endl;

return 0;

}

int average(int a, int b, int c) //Function header

{

int sum, avrg2;

sum = a + b + c;

avrg2 = sum / 3;

return avrg2; //Return statement

}

**QUESTION 5**

Find the errors in the following given code.

#include <iostream>

**//Error 1**

using namespace std;

int average(int, int); **//Error 2**

int power (float p); **//Error 3**

int main()

{

int x, y, z, avrg, powerOf;

cout << "Please enter three numbers:" << endl;

cin >> x >> y >> z;

avrg = average (); **//Error 4**

cout << "The average of the given three numbers is: " << avrg << endl;

power (); **//Error 5**

cout << "The average number to the power of two is: " << power () << endl; **//Error 6**

return 0;

}

int average(int a, int b, int c)

{

int sum, avrg2;

sum = a + b + c;

avrg2 = sum / 3;

**//Error 7**

}

int power (int p)

{

int pOf;

pOf = pow(p,2);

return 0; **//Error 8**

}

|  |  |
| --- | --- |
| Line that Contains Error | Corrected Statement |
| Missing <cmath> library reference | #include <cmath> |
| int average (int, int); | int average (int, int, int); |
| int power (float p); | int power (int p); |
| avrg = average (); | avrg = average (x,y,z); |
| power (); | powerOf = power (avrg); |
| cout << "The average number to the power of two is: " << power () << endl; | cout << "The average number to the power of two is: " << powerOf << endl; |
| No return statement for function average() | return avrg2; |
| return 0; | return pOf; |

**QUESTION 6**

Write a C++ program to calculate a rectangle’s area. The program consists of the following function:

* getLength – This function should ask the user to enter the rectangle’s length, and then returns that value as a double
* getWidth – This function should ask the user to enter the rectangle’s width, and then returns that value as a double.
* getArea – This function should accept the rectangle’s length and width as arguments and return the rectangle’s area.
* displayData – This function should accept the rectangle’s length, width and area as arguments, and display them in an appropriate message on the screen.
* main – This function consists of calls to the above functions.

For Question 6, provide the answer in .cpp file.

A black screen with text on it

Description automatically generated A screen shot of a computer

Description automatically generated

**Sample Output:**

**A black background with white text

Description automatically generated**