

Lab 06

Introduction to Modular Programming

**Muhammad Mujtaba**

**CMD ID: 540040**

[mmujtaba.bese25seecs@seecs.edu.pk](mailto:mmujtaba.bese25seecs@seecs.edu.pk)

**Class:** BESE 16B

**Batch:** 2k25

## Task 1 [CLO 1]:

### CODE:

#include <iostream>

#include <cmath>

int main()

{

// predefined constants

// will be optimzed away by compiler since constexpr

constexpr double PI = 3.14159;

constexpr double E = 2.71828;

// some functions defined in "cmath" header file

std::cout << "sqrt(4.0): " << std::sqrt(4.0) << "\n";

std::cout << "floor(-2.5: " << std::floor(-2.5) << "\n";

std::cout << "sin(2\*PI): " << std::sin(2 \* PI) << "\n";

std::cout << "asin(0.5): " << std::asin(0.5) << "\n";

std::cout << "cos(2\*PI): " << std::cos(2 \* PI) << "\n";

std::cout << "acos(0.5): " << std::acos(0.5) << "\n";

std::cout << "pow(2.0,2): " << std::pow(2.0, 2) << "\n";

std::cout << "atan(1.0): " << std::atan(1.0) << "\n";

std::cout << "log(E): " << std::log(E) << "\n";

std::cout << "ceil(-2.5): " << std::ceil(-2.5) << "\n";

std::cout << "exp(1.0): " << std::exp(1.0) << "\n";

std::cout << "floor(2.5): " << std::floor(2.5) << "\n";

std::cout << "max(2,std: " << std::max(2, std::min(3, 4)) << "\n";

std::cout << "log10(10.0): " << std::log10(10.0) << "\n";

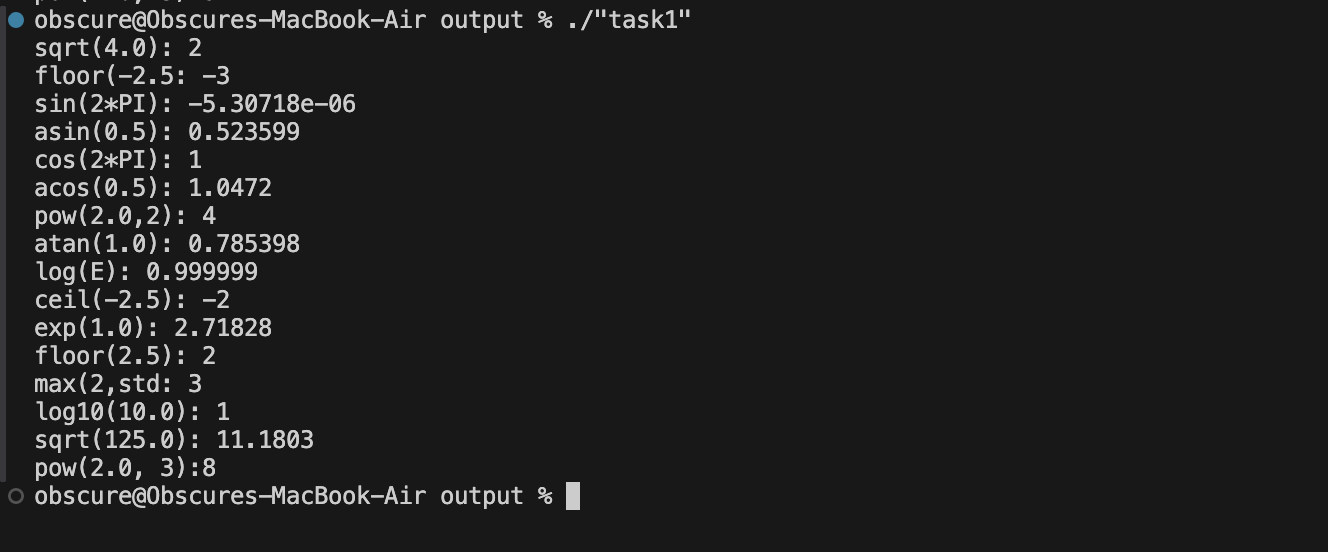
std::cout << "sqrt(125.0): " << std::sqrt(125.0) << "\n";

std::cout << "pow(2.0, 3):" << std::pow(2.0, 3) << "\n";

return 0;

}

### OUTPUT:



## Task 2 [CLO 2]:

### CODE:

#include <iostream>

#include <iomanip>

#include <math.h> // for M\_PI

#include <cmath>

double areaOfPentagon(double radius);

int main()

{

float number;

std::cout << "Enter the length from the centre to a vertex: ";

std::cin >> number;

std::cout << "The area of pentagon is " << std::fixed << std::setprecision(2) << areaOfPentagon(number) << "\n";

std::cin.ignore();

std::cin.get();

return 0;

}

double areaOfPentagon(double radius)

{

// M\_PI is from math.h

// get side from radius using formula

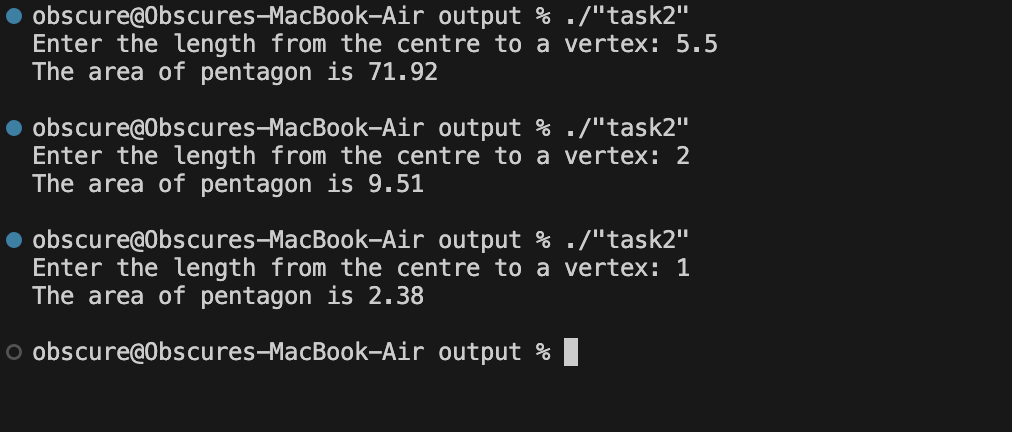
double side = 2 \* radius \* std::sin(M\_PI / 5);

// from calculate size, calculate and return area

return (5 \* std::pow(side, 2)) / (4 \* std::tan(M\_PI / 5));

}

### OUTPUT:



## Task 3 [CLO 1]:

### CODE:

#include <iostream>

#include <iomanip>

int main()

{

double monthlyPayment = 1345.4567;

double totalPayment = 866.887234;

std::cout << std::setprecision(7); // will not work; requieres fixed

std::cout << monthlyPayment << std::endl;

std::cout << totalPayment << std::endl;

std::cout << std::fixed << std::setprecision(2); // will work now

// all subsequent print statements printing float will be rounded down to two decimal places

std::cout << std::setw(8) << monthlyPayment << std::endl;

std::cout << std::setw(8) << totalPayment << std::endl;

std::cout << std::left; // all subsequent print statements will be aligned to left

std::cout << std::setw(8) << monthlyPayment << std::endl;

std::cout << std::setw(8) << totalPayment << std::endl;

std::cout << std::right; // all subsequent print statements will be aligned to right

std::cout << std::setw(8) << monthlyPayment << std::endl;

std::cout << std::setw(8) << totalPayment << std::endl;

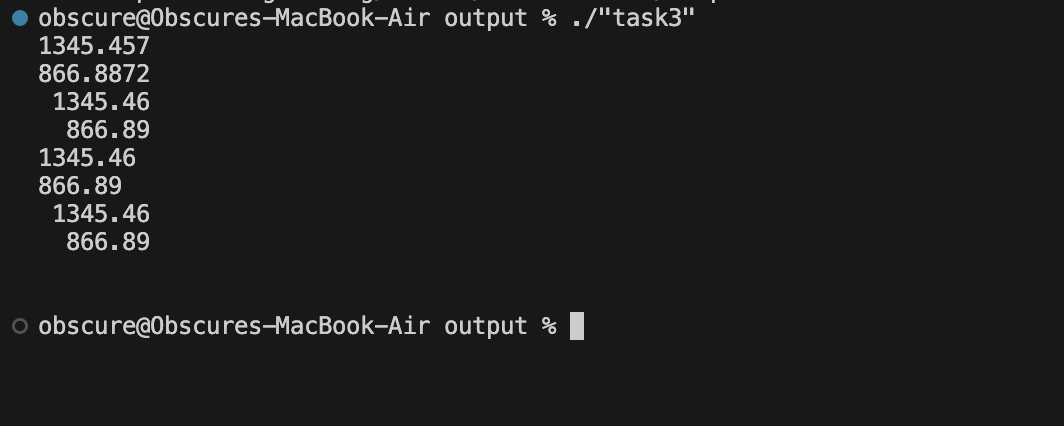
std::cin.ignore();

std::cin.get();

return 0;

}

### OUTPUT:



## Task 4 [CLO 2]:

### CODE:

#include <iostream>

#include <cstdlib>

#include <ctime>

int getRandom(int minInc, int maxInc);

bool askForGuess(int secret);

int main()

{

srand(time(nullptr)); // for seeding rand() function

// variables

int secret = getRandom(1, 100);

const size\_t max\_guesses = 5;

bool has\_guessed\_correctly = false;

std::cout << "Guess a magic number between 0 and 100.\n\n";

for (size\_t i = 0; i < max\_guesses; i++)

{

has\_guessed\_correctly = askForGuess(secret);

if (has\_guessed\_correctly)

break;

}

if (!has\_guessed\_correctly)

{

std::cout << "You could not guess the number, it was " << secret << ".\n\n";

}

// ignoring previous input

std::cin.ignore();

std::cin.get();

return 0;

}

// get random value for int (% operator only supports int) between given values inclusive

int getRandom(int minInc, int maxInc)

{

// we later subtract bigger from smaller so swapping

if (maxInc < minInc)

{

std::cerr << "WARNING> minInc is greater than maxInc. Swapping\n\n";

std::swap(minInc, maxInc);

}

// first term: calculate random number from 0 to (max - min): min will be added later

// second term: offset by min

return (rand() % maxInc - minInc + 1) + minInc;

}

bool askForGuess(int secret)

{

int guess;

std::cout << "Enter your guess: ";

std::cin >> guess;

// if guessed correctly then return true

if (guess == secret)

{

std::cout << "Yes, the number is " << secret << "\n\n";

return true;

}

else if (guess >= secret)

{

std::cout << "You guess is " << (abs((float)(guess - secret)) > 25 ? "too " : "") << "high" << "\n\n";

}

else if (guess <= secret)

{

std::cout << "You guess is " << (abs((float)(guess - secret)) > 25 ? "too " : "") << "low" << "\n\n";

}

return false;

}

### OUTPUT:

