**Austin Tesch**

**Chapter 5**

**Date: 05-16-24**

**Program 1: ch5\_pgm1; A program that asks the user for a positive integer value and computes the sum of all the integers from 1 up to the number entered.**

**Code:**

/\*

Programmer: Austin Tesch

Date modified: 05-15-24

Compiler used:  XCODE v. 15.0

Purpose:

A program that asks the user for a positive integer value and that uses a loop to validate the input.

The program should then use a second loop to compute the sum of all the integers from 1 up to the number entered.

For example, if the user enters 20, the loop will find the sum of 1+2+3+4+5+6+7+8+9+…17+18+19+20.

\*/

#include <iostream>

using namespace std;

int main (){

// Ask user for a positive integer value.

float input\_value;

cout << "Provide a positive integer: " << endl;

cin >> input\_value;

// Validate users input.

// If user input is not positive, ask again for positive integer.

while (input\_value <= 0)

{

cout << "It seems the number provided is not positive, please provide a positive integer: " << endl;

cin >> input\_value;

}

// User input is valid.

// Compute sum of all integers from 1 up to the users number.

int counter = 1, final\_sum = 0;

while (counter <= input\_value)

{

final\_sum += counter;

counter++;

}

// Return sum of integers to user.

cout << "The sum of all leading integers is: " << final\_sum << endl;

return 0;

}

**Output:**

**A black background with white text

Description automatically generated**

**Program 2: ch5\_pgm2; A program that creates a table showing the radius and area for a circle whose radius begins with 1 and continues doubling until it is 8.**

**Code:**

/\*

Programmer: Austin Tesch

Date modified: 05-15-24

Compiler used:  XCODE v. 15.0

Purpose:

A program that creates a table showing the radius and area for a circle whose radius begins with 1 and continues doubling until it is 8. Use 3.14 for PI.

The formula to compute the area of a circle is -- area = PI\*radius\*radius

So if a circle’s radius doubles (i.e., is multiplied by 2), the circle’s area will be four times as large as before.

\*/

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

int main () {

//Create circle with radius of 1.

const double PI = 3.14;

double radius = 1.0;

// Create table header.

cout << "RADIUS" << setw(15) << "AREA" << endl;

cout << "----------------------" << endl;

// Populate table with area and radius for circles until radius is 8.

double area;

// Increment radius and ouput to table until radius is 8.

while (radius <= 8)

{

area = PI \* sqrt(radius);

cout << radius << setw(20) << setprecision(3) << area << endl;

radius \*= 2;

}

return 0;

}

**Output:**

**A black and white text on a black background

Description automatically generated**

**Program 3: ch5\_pgm3; A program that generates a random number between 1 and 100 and asks the user to guess what the number is.**

**Code:**

/\*

Programmer: Austin Tesch

Date modified: 05-16-24

Compiler used:  XCODE v. 15.0

Purpose:

A program that generates a random number between 1 and 100 and asks the user to guess what the number is.

- If the user’s guess is higher than the random number, the program should display “Too high. Try again.”

- If the user’s guess is lower than the random number, the program should display “Too low. Try again.”

- The program should use a loop that repeats until the user correctly guesses the random number.

- Then the program should display “Congratulations. You figured out my number.”

\*/

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main () {

// Use time fucntion to create "seed" for random number generation.

srand(static\_cast <unsigned int> (time(NULL)));

// Set MAXIMUM AND MINIMUM

int const MIN\_VALUE = 1; // Minimum value is probably not nessesary for this application.

int const MAX\_VALUE = 100;

// Generate a random number

int random\_number = rand() % (MAX\_VALUE - MIN\_VALUE + 1) + MIN\_VALUE;

// Prompt user to guess the random number

int user\_input;

cout << "Welcome to the number guessing game! \n"

<< "I've guessed a number between 1 and 100. \n"

<< "Can you guess it?" << endl;

cin >> user\_input;

// Validate users number with random number

// If the user’s guess is higher than the random number, display “Too high. Try again.”

// If the user’s guess is lower than the random number, display “Too low. Try again.”

while (user\_input != random\_number)

{

if (user\_input > random\_number) {

cout << "Too high. Try again." << endl;

cin >> user\_input;

} else {

cout << "Too low. Try again." << endl;

cin >> user\_input;

}

}

// Once user has guessed the correct number, display “Congratulations. You figured out my number.”

cout << "Congratulations. You figured out my number." << endl;

return 0;

}

**Output:**

****

**Program 4: ch5\_pgm4; A program that calculates the occupancy rate of the 120 suites (20 per floor) located on the top six floors of a 15-story luxury hotel.**

**Code:**

/\*

Programmer: Austin Tesch

Date modified: 05-16-24

Compiler used:  XCODE v. 15.0

Purpose:

A program that calculates the occupancy rate of the 120 suites (20 per floor) located on the top six floors of a 15-story luxury hotel.

These are floors 10–12 and 14–16 because, like many hotels, there is no 13th floor.

Solve the problem by using a single loop that iterates once for each floor between 10 and 16 and, on each iteration, asks the user to input the number of suites occupied on that floor.

Use a nested loop to validate that the value entered is between 0 and 20.

After all the iterations, the program should display how many suites the hotel has, how many of them are occupied, and what percentage of them are occupied.

\*/

#include <iostream>

using namespace std;

int main(){

// Set floor values.

int lowest\_floor = 10, highest\_floor = 16;

int total\_floors = highest\_floor - lowest\_floor;

int const SUITES\_PER\_FLOOR = 20;

int const TOTAL\_SUITES = total\_floors \* SUITES\_PER\_FLOOR;

int occupied\_suites = 0;

// Iterate thorugh floors.

int floor, user\_input;

for (floor = lowest\_floor; floor <= highest\_floor; ++floor){

// No 13th floor, skip.

if (floor == 13)

continue;

do {

// Prompt user for total number of occupied floors.

cout << "Enter the number of occupied suites on floor " << floor << ": " << endl;

cin >> user\_input;

// Validate input.

if (user\_input < 0 || user\_input > 20){

cout << "Input invalid." << endl;

}

} while (user\_input < 0 || user\_input > 20);

// User supplied valid input, move to next floor.

occupied\_suites += user\_input;

}

// Display total number of suites avaliable

int suites\_avaliable = TOTAL\_SUITES - occupied\_suites;

cout << "The top 5 floors of the hotel has " << TOTAL\_SUITES << " avaliable suites." << endl;

// Display percentage of occupied suites

double (occupancy\_rate) = (static\_cast<double>(occupied\_suites) / TOTAL\_SUITES) \* 100.0;

cout << occupancy\_rate << "% of the top floor suites are unavaliable." << endl;

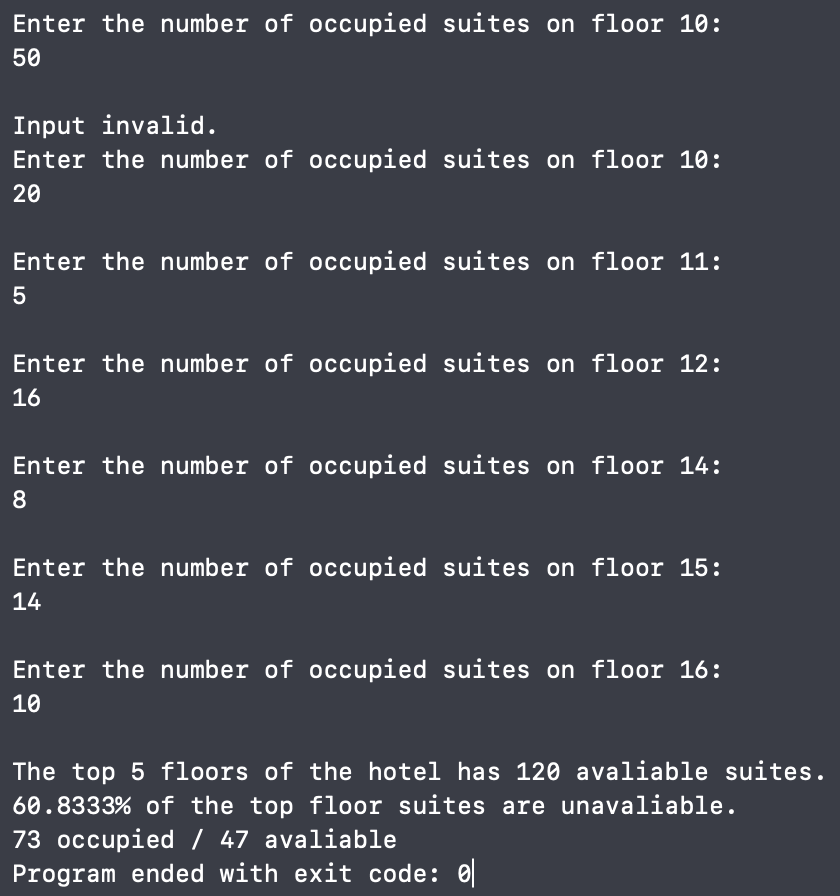
// Display occupied and unoccupied numbers.

cout << occupied\_suites << " occupied / " << suites\_avaliable << " avaliable" << endl;

return 0;

}

**Output:**

****