1. Find the errors in the following overloaded functions.

int getValue()

{

int inputValue;

cout << "Enter an integer: ";

cin >> inputValue;

return inputValue;

}

double getValue()

{

double inputValue;

cout << "Enter a floating-point number: ";

cin >> inputValue;

return inputValue;

}

-> This is not a case of overloaded functions. The functions have different return types instead of paramenter.

1. Celsius Temperature Table

The formula for converting a temperature from Fahrenheit to Celsius is

C = (5/9)\*(F−32)

where F is the Fahrenheit temperature and C is the Celsius temperature. Write a function named celsius that accepts a Fahrenheit temperature as an argument. The function should return the temperature, converted to Celsius. Demonstrate the function by calling it in a loop that displays a table of the Fahrenheit temperatures 0 through 20 and their Celsius equivalents.

#include <iostream>

using namespace std;

// Convert fahrenheit to celsius

double fto(double f) {

double c = (5.0f / 9.0f) \* (f - 32.0f);

return c;

}

int main() {

for (int f = 0; f <= 20; f++) {

printf("%dF --> %.2lfC \n", f, ftoc(f));

}

return 0;

}

1. Coin Toss

Write a function named coinToss that simulates the tossing of a coin. When you call the function, it should generate a random number in the range of 1 through 2. If the random number is 1, the function should display “heads.” If the random number is 2, the function should display “tails.” Demonstrate the function in a program that asks the user how many times the coin should be tossed and then simulates the tossing of the coin that number of times.

#include <iostream>

#include <random>

#include <ctime>

using namespace std;

void coin\_toss() {

random\_device rdev;

uniform\_int\_distribution<int>dist(1,2);

if (dist(rdev) == 1) {

cout << "Head" << endl;

} else if (dist(rdev) == 2){

cout << "Tails" << endl;

}

}

int main() {

cout << "Enter the number of times the coin will be tossed: ";

int toss = 0;

cin >> toss;

// While the user does not input a number greater than or 0

while (toss < -1) {

cout << "Invalid input.\r" << endl;

cin >> toss;

}

// If the user does not input any toss count

if (toss == 0) {

cout << "The coin was not tossed." << endl;

return 0;

}

// Toss the coin.

int counter = 1;

while (counter <= toss) {

coin\_toss();

counter++;

}

return 0;

}

1. Days Out

Write a program that calculates the average number of days a company’s employees are absent. The program should have the following functions:

A function called by main that asks the user for the number of employees in the company. This value should be returned as an int. (The function accepts no arguments.)

A function called by main that accepts one argument: the number of employees in the company. The function should ask the user to enter the number of days each employee missed during the past year. The total of these days should be returned as an int.

A function called by main that takes two arguments: the number of employees in the company and the total number of days absent for all employees during the year. The function should return, as a double, the average number of days absent. (This function does not perform screen output and does not ask the user for input.)

#include <iostream>

#include <cstdlib>

using namespace std;

int get\_emp(){

cout << "Enter the number of employees in the company --> ";

int emp\_count = 0;

cin >> emp\_count;

if (emp\_count <= 0) {

if (emp\_count == 0) {

cout << "You have enter no employees" << endl;

exit(1);

}

else {

cout << "The count of employees cannot be less than 0. " << endl;

exit(1);

}

}

return emp\_count;

}

int missed\_days(int emps\_count){

int total\_d = 0;

for (int e = 1; e <= emps\_count; e++){

cout << "Enter the number of missed days for employee: " << e << endl;

int d = 0;

cin >> d;

int tries = 3;

while (d < 0){

tries --;

cout << "The entered day cannot be less than 0" << endl;

if (tries <= 0){

cout << "The program has terminated." << endl;

exit(EXIT\_SUCCESS);

}

cin >> d;

}

total\_d += d;

}

return total\_d;

}

double get\_d\_aver(int emps, int d\_total){

double average = static\_cast<double>(d\_total/emps);

return average;

}

int main(){

int employee\_count = get\_emp();

int mi\_days = missed\_days(employee\_count);

cout << "The average of missed days between all employees is: "

<< get\_d\_aver(employee\_count, mi\_days) << endl;

return 0;

}

1. Rectangle Area

Write a the program that will ask the user to enter the width and length of a rectangle and then display the rectangle’s area. The program calls the following functions:

getLength – This function should ask the user to enter the rectangle’s length and then return that value as a double.

getWidth – This function should ask the user to enter the rectangle’s width and then return that value as a double.

getArea – This function should accept the rectangle’s length and width as arguments and return the rectangle’s area. The area is calculated by multiplying the length by the width.

displayData – This function should accept the rectangle’s length, width, and area as arguments and display them in an appropriate message on the screen.

#include <iostream>

#include <random>

#include <ctime>

using namespace std;

#include <iostream>

#include <random>

#include <ctime>

using namespace std;

struct rectangle {

double length = 1.0f;

double width = 1.0f;

double area = 1.0f;

} rectangle1;

double get\_length();

double get\_width();

double get\_area();

void display\_data(double, double, double);

int main() {

display\_data(get\_area(), get\_length(), get\_width());

return 0;

}

double get\_length() {

cout << "Enter the length of the rectangle: ";

cin >> rectangle1.length;

if (rectangle1.length <= 0.0f){

cout << "The length cannot be less than 0 " << endl;

rectangle1.length = 1.0f;

}

return rectangle1.length;

}

double get\_width() {

cout << "Enter the width of the rectangle: ";

cin >> rectangle1.width;

if (rectangle1.width <= 0.0f){

cout << "The width cannot be less than 0 " << endl;

rectangle1.width = 1.0f;

}

return rectangle1.width;

}

double get\_area() {

return rectangle1.length \* rectangle1.width;

}

void display\_data(double area, double width, double length) {

cout << "Rectangle Aspects:: " << endl;

cout << "\tLength--> " << length << endl;

cout << "\twidth--> " << width << endl;

cout << "\tArea--> " << area << endl;

}