**C++ Part I (INFO1-CE9264) Fall 2014 – Quiz 1 and Homework 3**

Clement Chan

**Quiz 1**

#include<iostream>

#include<cmath>

using namespace std;

const float R=10;

const double D=20;

const float H=20;

const double PI = 3.14159265359;

//Class that define all the variables and methods

class CylinderCalc{

private:

float radius;

float height;

double diameter;

public:

float SurfaceArea(float,float);

float SurfaceArea(double,double);

float Perimeter\_Cylinder(float,float);

float Perimeter\_Cylinder(double,float);

float Volume\_Cylinder(float,float);

float Volume\_Cylinder(double,float);

};

//Class Methods Definitions

float CylinderCalc::SurfaceArea(float r, float h){

radius = r;

height = h;

float Top\_Bottom = 2 \* PI \* radius \* radius;

float circumference = 2 \* PI \* radius;

float cylinderarea = circumference \* height;

float Total\_Surfacearea = Top\_Bottom + cylinderarea;

return Total\_Surfacearea;

}

float CylinderCalc::SurfaceArea(double d, double h=10){

diameter = d;

height = h;

float Top\_Bottom = 2 \* PI \* pow(diameter/2,2.0);

float circumference = 2 \* PI \* (diameter/2);

float cylinderarea = circumference \* height;

float Total\_Surfacearea = Top\_Bottom + cylinderarea;

return Total\_Surfacearea;

}

float CylinderCalc::Perimeter\_Cylinder(float r, float h){

radius = r;

height = h;

float perimeter\_circles = 2\*(2\*PI\*radius);

float perimeter\_body = (2\*PI\*radius + height)\*2;

float perimeter\_total = perimeter\_circles + perimeter\_body;

return perimeter\_total;

}

float CylinderCalc::Perimeter\_Cylinder(double d, float h){

diameter = d;

height = h;

float perimeter\_circles = 2\*(2\*PI\*(diameter/2));

float perimeter\_body = (2\*PI\*(diameter/2) + height)\*2;

float perimeter\_total = perimeter\_circles + perimeter\_body;

return perimeter\_total;

}

float CylinderCalc::Volume\_Cylinder(float r, float h){

radius = r;

height = h;

float volume = PI \* radius \* radius \* height;

return volume;

}

float CylinderCalc::Volume\_Cylinder(double d, float h){

diameter = d;

height = h;

float volume = PI \* pow(diameter/2,2.0) \* height;

return volume;

}

int main(){

CylinderCalc CC;

cout<<"Surface area with float r and float h is: "<<CC.SurfaceArea(R,H)<<endl;

cout<<"Surface area with double d and double h=10 is: "<<CC.SurfaceArea(D)<<endl;

cout<<"Perimeter with float r and float h is: "<<CC.Perimeter\_Cylinder(R,H)<<endl;

cout<<"Perimeter with double d and float h is: "<<CC.Perimeter\_Cylinder(D,H)<<endl;

cout<<"Volume with float r and float h is: "<<CC.Volume\_Cylinder(R,H)<<endl;

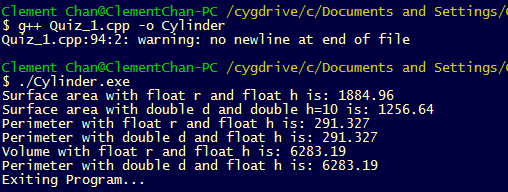
cout<<"Perimeter with double d and float h is: "<<CC.Volume\_Cylinder(D,H)<<endl;

cout<<"Exiting Program..."<<endl;

return 0;

}

**Output:**

****

**Homework 3**

**Question 1**

#include<iostream>

#include<cmath>

#include<cstdio>

using namespace std;

//Function Prototypes

void InputTime(int &hours, int &minutes);

char Day\_Night(int &hours, int &minutes);

void TimeConversion(int &hours, int &minutes);

void Output(int &hours, int &minutes, char &am\_pm);

//Main Class

int main(){

int hrs, minutes;

char am\_pm;

char ans;

do{

InputTime(hrs, minutes);

am\_pm = Day\_Night(hrs, minutes);

TimeConversion(hrs, minutes);

Output(hrs, minutes, am\_pm);

cout<<"Do you want to restart (y/n)?"<<endl;

cin >> ans;

}while(ans == 'Y' || ans == 'y');

cout<<"Exiting Program..."<<endl;

return 0;

}

//Function Definitions

void InputTime(int &hrs, int &minutes)

{

cout << "Enter hours in 24 hours format: "<<endl;

cin >> hrs;

if(hrs > 24 || hrs < 0){

printf("Hours not in 24 hours range.");

exit(0);

}

cout << "Enter minutes: " <<endl;

cin >> minutes;

if(minutes > 60 || minutes < 0){

printf("Minutes not in 60 minutes range.");

exit(0);

}

}

char Day\_Night (int &hrs, int &minutes)

{

char am\_pm;

if(hrs >=12 && hrs < 24){

am\_pm = 'P';

}

else{

am\_pm = 'A';

}

return am\_pm;

}

void TimeConversion(int &hrs, int &minutes)

{

if (hrs >= 12){

hrs = hrs - 12;

}

else{

hrs = hrs;

}

}

//Output of Day\_Night

void Output(int &hours, int &minutes, char &am\_pm)

{

if (minutes < 10){

cout << hours << ":0" << minutes << " " << am\_pm << endl;

}

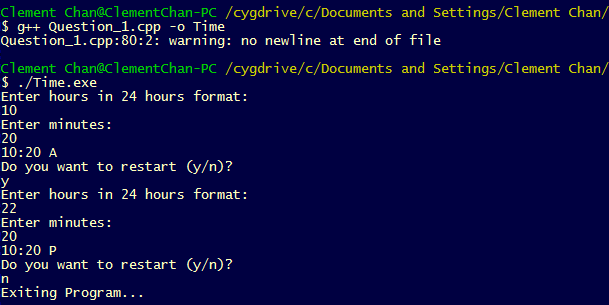
else{

cout << hours << ":" << minutes << " " << am\_pm << endl;

}

}

**Output:**



**Question 4**

#include<iostream>

#include<cmath>

#include<cstdio>

#include<iomanip>

using namespace std;

const float meters\_in\_foot = 0.3048;

const float cm\_in\_m = 100;

const float inches\_in\_feet = 12;

//Function Prototypes

void InputHeight(float &feet, float &inches);

void ConvertHeight(float &feet, float &inches, float &meters, float &centimeters);

void Output(float &feet, float &inches, float &meters , float &centimeters);

//Main Class

int main(){

float fts, inch, meters, centimeters;

char ans;

do{

InputHeight(fts, inch);

ConvertHeight(fts, inch, meters, centimeters);

Output(fts, inch, meters, centimeters);

cout<<"Do you want to continue (y/n)?"<<endl;

cin >> ans;

} while (ans == 'Y' || ans == 'y');

cout<<"Exiting Program..."<<endl;

return 0;

}

//Function Definitions

void InputHeight(float &feet, float &inches)

{

cout << "Enter feet: "<<endl;

cin >> feet;

if (feet < 0){

printf("feet entered exceed value.");

exit(0);

}

cout << "Enter inches: "<<endl;

cin >> inches;

if (inches > 12 || inches < 0){

printf("inches entered exceed value.");

exit(0);

}

}

void ConvertHeight(float &feet, float &inches, float &meters, float &centimeters)

{

float i\_to\_f;

i\_to\_f = inches / (inches\_in\_feet) \* meters\_in\_foot;

meters = (feet \* meters\_in\_foot) + i\_to\_f;

centimeters = (meters - floor(meters)) \* 100;

meters = floor(meters);

}

//Output of Day\_Night

void Output(float &feet, float &inches, float &meters , float &centimeters)

{

cout<< "From " << feet << " feet and " << inches << " inches to " << endl;

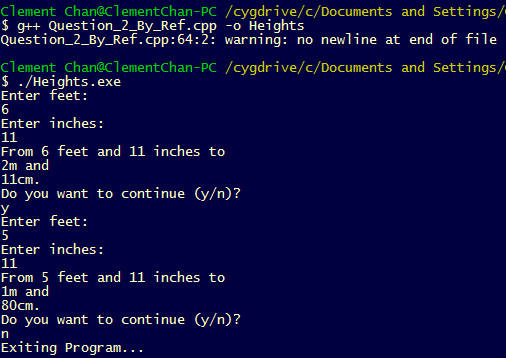
cout << meters << "m and "<<endl;

cout.precision(2);

cout << centimeters << "cm." << endl;

}

**Output:**



**Question 7**

#include<iostream>

#include<cmath>

#include<cstdio>

#include<iomanip>

using namespace std;

const float pounds\_in\_kilos = 2.2046;

const float g\_in\_kg = 1000;

const float ounces\_in\_pound = 16;

//Function Prototypes

void InputWeight(float &pounds, float &ounces);

void ConvertWeight(float &pounds, float &ounces, float &kilos, float &grams);

void Output(float &pounds, float &ounces, float &kilos, float &grams);

//Main Class

int main(){

float pounds, ounces, kilos, grams;

char ans;

do{

InputWeight(pounds, ounces);

ConvertWeight(pounds, ounces, kilos, grams);

Output(pounds, ounces, kilos, grams);

cout<<"Do you want to continue (y/n)?"<<endl;

cin >> ans;

} while (ans == 'Y' || ans == 'y');

cout<<"Exiting Program..."<<endl;

return 0;

}

//Function Definitions

void InputWeight(float &pounds, float &ounces)

{

cout << "Enter Pounds: "<<endl;

cin >> pounds;

cout << "Enter Ounces: "<<endl;

cin >> ounces;

if (pounds < 0 || ounces > 16 || ounces < 0){

printf("Inputs entered exceed value.");

exit(0);

}

}

void ConvertWeight(float &pounds, float &ounces, float &kilos, float &grams)

{

float o\_to\_k;

o\_to\_k =(ounces / (ounces\_in\_pound)) / pounds\_in\_kilos;

kilos = (pounds / pounds\_in\_kilos) + o\_to\_k;

grams = (kilos-floor(kilos));

grams = grams\*1000;

}

//Output of Day\_Night

void Output(float &pounds, float &ounces, float &kilos, float &grams)

{

cout<< "From " << pounds << " pounds and " << ounces << " ounces to " << endl;

cout.precision(2);

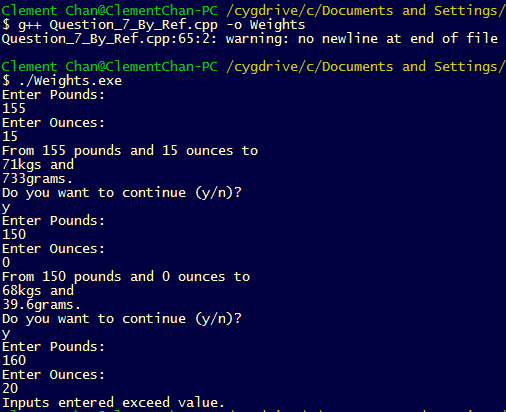
cout<< kilos << "kgs and " << endl;

cout.precision(3);

cout<< grams << "grams." << endl;

}

**Output:**

****

**Question 15**

#include<iostream>

#include<cmath>

#include<cstdio>

#include<iomanip>

using namespace std;

//Function Prototypes

void GCD(int &numerator, int &denominator, int &gcdvalue);

void ConvertToLowest(int &numerator, int &denominator, int &gcdvalue);

void Output(int &numerator, int &denominator, int &gcdvalue);

//Main Class

int main(){

int numerator, denominator;

int gcdvalue = 0;

GCD(numerator, denominator, gcdvalue);

ConvertToLowest(numerator, denominator, gcdvalue);

Output(numerator, denominator, gcdvalue);

cout<<"Exiting Program..."<<endl;

return 0;

}

//Function Definitions

void GCD(int &numerator, int &denominator, int &gcdvalue)

{

//variables definition

int n, nmod, dmod;

bool CheckValue;

cout << "Enter numerator: "<<endl;

cin >> numerator;

cout << "Enter denominator: "<<endl;

cin >> denominator;

if(denominator == 0){

CheckValue = true;

printf("Denominator cannot be zero.");

exit(0);

}

else

{

CheckValue = false;

}

if(CheckValue == 0 && numerator > denominator){

n = numerator;

}

if(CheckValue == 0 && numerator < denominator){

n = denominator;

}

for (int i; i<=n; i++){

nmod = numerator % i;

dmod = denominator % i;

if(nmod == 0 && dmod == 0){

gcdvalue = i;

}

if(i == n && gcdvalue == 0){

printf("There is no common divisor between numerator and denominator...");

exit(0);

}

}

}

void ConvertToLowest(int &numerator, int &denominator, int &gcdvalue)

{

numerator = numerator/gcdvalue;

denominator = denominator/gcdvalue;

}

//Output Function

void Output(int &numerator, int &denominator, int &gcdvalues)

{

cout << "Numerator (Lowest Converted) Output: " << numerator << endl;

cout << "Denominator (Lowest Convereted) Output: " << denominator << endl;

cout << "GCD value is: " << gcdvalues << endl;

}

**Output:**

