**C++ Part I (INFO1-CE9264) Fall 2014 – Homework 2**

Clement Chan

**Question 1:**

**Code:**

#include<iostream>

using namespace std;

//Universal data type

const double liters\_to\_gallons = 0.264179;

//Function Declaration

double miles\_per\_hour (double liters, double miles);

//Method function

double miles\_per\_hour (double liters, double miles){

double mph;

mph = (liters \* liters\_to\_gallons) / miles;

return mph;

}

int main(){

double liters;

double miles;

char ans;

do{

cout<<"Please enter the liters of gasoline consumed by the car: " <<endl;

cin>>liters;

cout<<"Please enter the miles that the car has traveled: " << endl;

cin>>miles;

cout << "Miles per hour value is: " << miles\_per\_hour(liters, miles);

cout<<" mph.\n";

cout<<"Again (y/n)?: ";

cin>>ans;

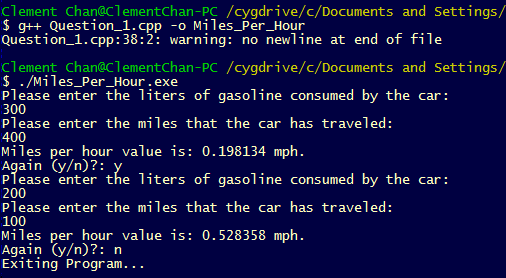
} while (ans =='y' || ans == 'Y');

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

return 0;

}

**Output:**

****

**Question 2:**

**Code:**

#include<iostream>

using namespace std;

//Function Declaration

double inflation(double priceprevious, double pricenow);

//Function

double inflation(double priceprevious, double pricenow){

double infrate;

infrate = (pricenow - priceprevious) / priceprevious \* 100;

return infrate;

}

int main(){

double priceprevious;

double pricenow;

char ans;

do {

cout << "Please Enter the price of the item last year: " << endl;

cin >> priceprevious;

cout << "Please Enter the price of the item this year: " << endl;

cin >> pricenow;

cout << "The inflation is :" << inflation(priceprevious, pricenow);

cout << "%. \n";

cout << "Repeat program? (y/n)" << endl;

cin >> ans;

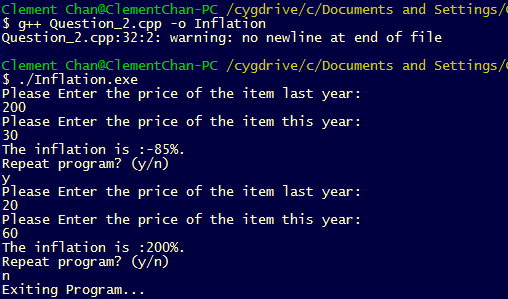
} while (ans =='y' || ans == 'Y');

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

return 0;

}

**Output:**

****

**Question 4:**

**Code:**

#include<iostream>

#include<cmath>

using namespace std;

const double gconstant= 0.00000006673;

//Function Declaration

double gforce(double m1, double m2, double distance);

//Function

double gforce(double m1, double m2, double distance){

double gravityforce;

gravityforce = (gconstant \* m1 \* m2) / (pow(distance, 2.0));

return gravityforce;

}

int main(){

double mass\_1;

double mass\_2;

double distance;

char ans;

do {

cout << "Please Enter the first mass in grams (g): " << endl;

cin >> mass\_1;

cout << "Please Enter the second mass in grams (g): " << endl;

cin >> mass\_2;

cout << "Please Enter the distance in centimeters (cm): " << endl;

cin >> distance;

cout << "The Gravity Force in dynes: " << gforce(mass\_1, mass\_2, distance);

cout << " [g\*cm/(s\*s)] \n";

cout << "Repeat program? (y/n)" << endl;

cin >> ans;

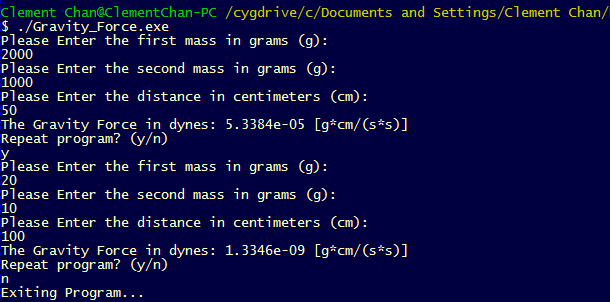
} while (ans =='y' || ans == 'Y');

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

return 0;

}

**Output:**

****

**Question 6:**

**Code:**

#include<iostream>

#include<cmath>

using namespace std;

//Function Declaration

double averagescores(double s1, double s2, double s3, double s4);

double stdscores(double s1, double s2, double s3, double s4, double avescore);

//Functions

double averagescores(double s1, double s2, double s3, double s4){

double avescore;

avescore = (s1 + s2 + s3 + s4) / 4;

return avescore;

}

double stdscores(double s1, double s2, double s3, double s4, double avescore){

double stds;

stds = sqrt(pow((s1-avescore),2.0) + pow((s2-avescore),2.0) + pow((s3-avescore),2.0) + pow((s4-avescore),2.0));

return stds;

}

int main(){

double score\_1;

double score\_2;

double score\_3;

double score\_4;

double averagescore;

char ans;

do {

cout << "Please Enter the first score: " << endl;

cin >> score\_1 ;

cout << "Please Enter the second score: " << endl;

cin >> score\_2 ;

cout << "Please Enter the third score: " << endl;

cin >> score\_3 ;

cout << "Please Enter the fourth score: " << endl;

cin >> score\_4 ;

averagescore = averagescores(score\_1, score\_2, score\_3, score\_4);

cout << "The average score of the 4 scores input is: " << averagescore ;

cout << "\n";

cout << "The standard deviation of the 4 scores input is: " << stdscores(score\_1, score\_2, score\_3, score\_4, averagescore);

cout << "\n";

cout << "Repeat program? (y/n)" << endl;

cin >> ans;

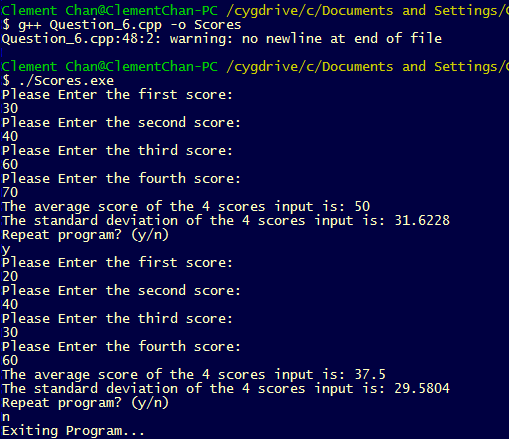
} while (ans =='y' || ans == 'Y');

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

return 0;

}

**Output:**

****

**Question 12:**

**Code:**

#include <iostream>

#include <stdio.h>

#include <cmath>

using namespace std;

//Function Declarations

bool isLeapYear(int year);

int getCenturyValue(int year);

int getYearValue(int year);

bool CheckDates (int month, int day);

int getMonthValue(int month, int year);

char\* getDayvalue (int day, int monthvalue, int yearvalue, int centuryvalue);

//Functions

bool isLeapYear(int year){

bool Bvalue;

if ((year % 400 == 0) || ((year % 4 == 0) && (year % 100 != 0))){

return Bvalue = true;

}

else{

return Bvalue = false;

}

}

int getCenturyValue(int year){

int century = year / 100;

int remainder = century % 4;

int finalcenturyvalue = (3-remainder)\*2;

return finalcenturyvalue;

}

int getYearValue(int year){

int lasttwodigit = year % 100;

int b = lasttwodigit / 4;

int finalyearvalue = lasttwodigit + b;

return finalyearvalue;

}

enum MonthLength{

JAN\_LENGTH = 31, FEB\_LENGTH = 28, MAR\_LENGTH = 31,

APR\_LENGTH = 30, MAY\_LENGTH = 31, JUN\_LENGTH = 30,

JULY\_LENGTH = 31, AUG\_LENGTH = 31, SEP\_LENGTH = 30,

OCT\_LENGTH = 31, NOV\_LENGTH = 30, DEC\_LENGTH = 31

};

bool CheckDates (int month, int day){

bool RValue;

if (month = 1 && day > JAN\_LENGTH){

return RValue = true;

}

if (month = 2 && day > FEB\_LENGTH){

return RValue = true;

}

if (month = 3 && day > MAR\_LENGTH){

return RValue = true;

}

if (month = 4 && day > APR\_LENGTH){

return RValue = true;

}

if (month = 5 && day > MAY\_LENGTH){

return RValue = true;

}

if (month = 6 && day > JUN\_LENGTH){

return RValue = true;

}

if (month = 7 && day > JULY\_LENGTH){

return RValue = true;

}

if (month = 8 && day > AUG\_LENGTH){

return RValue = true;

}

if (month = 9 && day > SEP\_LENGTH){

return RValue = true;

}

if (month = 10 && day > OCT\_LENGTH){

return RValue = true;

}

if (month = 11 && day > NOV\_LENGTH){

return RValue = true;

}

if (month = 12 && day > DEC\_LENGTH){

return RValue = true;

}

}

int getMonthValue(int month, int year){

int monthvalue;

if(isLeapYear(year) == 1){

if(month == 1){

return monthvalue = 6;

}

if(month == 2){

return monthvalue = 2;

}

}

if(isLeapYear(year) == 0){

if(month == 1){

return monthvalue = 0;

}

if(month == 2){

return monthvalue = 3;

}

}

if(month == 3){

return monthvalue = 3;

}

if(month == 4){

return monthvalue = 6;

}

if(month == 5){

return monthvalue = 1;

}

if(month == 6){

return monthvalue = 4;

}

if(month == 7){

return monthvalue = 6;

}

if(month == 8){

return monthvalue = 2;

}

if(month == 9){

return monthvalue = 5;

}

if(month == 10){

return monthvalue = 0;

}

if(month == 11){

return monthvalue = 3;

}

if(month == 12){

return monthvalue = 5;

}

}

char\* getDayvalue (int day, int monthvalue, int yearvalue, int centuryvalue){

char\* DayName;

int corresponding = day + monthvalue + yearvalue + centuryvalue;

int remainder = corresponding % 7;

if (remainder == 0){

return DayName = "Sunday";

}

if (remainder == 1){

return DayName = " Monday";

}

if (remainder == 2){

return DayName = "Tuesday";

}

if (remainder == 3){

return DayName = "Wednesday";

}

if (remainder == 4){

return DayName = "Thursday";

}

if (remainder == 5){

return DayName = "Friday";

}

if (remainder == 6){

return DayName = "Saturday";

}

}

int main(){

int year;

int month;

int day;

char ans;

cout << "Please enter the year: " << endl;

cin >> year;

cout << "Please enter month: " << endl;

cin >> month;

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

printf("Month entered exceed limit. Please re-enter.");

return 0;

}

cout << "Please enter day: " << endl;

cin >> day;

if (CheckDates (month, day) == 1){

printf("Day entered exceeds maximum date. Please Re-enter date.\n");

return 0;

}

else {

cout<< "The Day is: " << getDayvalue (day, getMonthValue(month, year), getYearValue(year), getCenturyValue(year));

cout<< "\n";

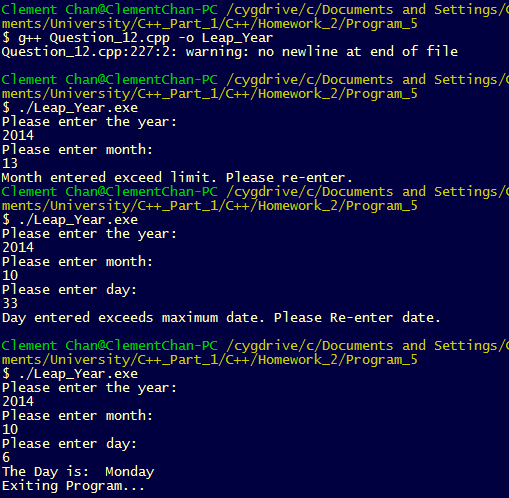
cout<<"Exiting Program..."<<endl;

return 0;

}

}

**Output:**

****