|  |  |
| --- | --- |
| **Name** | **Nazeer ahmed** |
| **CMS ID** | **023-18-0062** |
| **Marks** |  |
| **Date** |  |

**Objective:** To become familiar with user defined functions

A function is a group of statements that together perform a task. Every C++ program has at least one function, which is main(), and all the most trivial programs can define additional functions. You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division usually is so each function performs a specific task. A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function. The C++ standard library provides numerous built-in functions that your program can call. For example, function strcat() to concatenate two strings, function memcpy() to copy one memory location to another location and many more functions.

Defining a Function:

The general form of a C++ function definition is as follows:

return\_type function\_name( parameter list )

{

body of the function

}

A C++ function definition consists of a function header and a function body. Here are all the parts of a function:

**Return Type:** A function may return a value. The return\_type is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return\_type is the keyword void.

**Function Name:** This is the actual name of the function. The function name and the parameter list together constitute the function signature.

**Parameters:** A parameter is like a placeholder. When a function is invoked, you pass a value to the parameter. This value is referred to as actual parameter or argument. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters.

**Function Body:** The function body contains a collection of statements that define what the function does.

**Function Declarations:**

A function declaration tells the compiler about a function name and how to call the function. The actual body of the function can be defined separately.

A function declaration has the following parts:

return\_type function\_name( parameter list );

Calling a Function:

While creating a C++ function, you give a definition of what the function has to do. To use a function, you will have to call or invoke that function.

When a program calls a function, program control is transferred to the called function. A called function performs defined task and when its return statement is executed or when its function-ending closing brace is reached, it returns program control back to the main program

**Program# 1:**

#include <iostream>

using namespace std;

// function declaration

int max(int num1, int num2);

int main ()

{

// local variable declaration:

int a = 100;

int b = 200;

int ret;

// calling a function to get max value.

ret = max(a, b);

cout << "Max value is : " << ret << endl;

return 0;

}

// function returning the max between two numbers

int max(int num1, int num2)

{

// local variable declaration

int result;

if (num1 > num2)

result = num1;

else

result = num2;

return result;

}

|  |
| --- |
| Output: |

**Program#2:**

Following program uses S=VT formula to calculate distance, Velocity and Time. In this program three functions are used i-e one for distance second for velocity and third for time.

#include<iostream>

using namespace std;

double dist(double v, double t){

double dist=v\*t;

return dist;

}

double vel(double s, double t){

double vel=s/t;

return vel;

}

double tim(double v, double s){

double tim=s/v;

return tim;

}

int main(){

double s,v,t;

int x;

cout<<"This program can calculate distance, time and velocity:"<<endl;

cout<<"Enter 1 to calculate distance:"<<endl;

cout<<"Enter 2 to calculate Velocity:"<<endl;

cout<<"Enter 3 to calculate Time:"<<endl;

cin>>x;

switch(x){

case 1:

cout<<"Enter the value of velocity and time:"<<endl;

cin>>s,t;

cout<<"The Velocity is:"<<dist(s,t)<<endl;

break;

case 2:

cout<<"Enter the value of distance and time:"<<endl;

cin>>s,t;

cout<<"The distance is:"<<vel(v,t)<<endl;

break;

case 3:

cout<<"Enter the value of distance and velocity:"<<endl;

cin>>s,v;

cout<<"The Time is:"<<tim(v,s)<<endl;

break;

default:

cout<<"Values are incorrect:"<<endl;

}

return 0;

}

**Program#3:**

*#include <iostream>*

*using* *namespace* std;

*void* duplicate (*int* a, *int* b, *int* c)

{

a\*=2;

b\*=2;

c\*=2;

cout << "x=" << x << ", y=" << y << ", z=" << z; }

int main ()

{

*int* x=1, y=3, z=7;

duplicate (x, y, z);

return 0;

}

**Programing Exercise:**

1. Write a C++ program which defines four void functions final\_velocity, initial\_velocity, Distance and acceleration, the program uses the formula

2as=(vf2-vi2)/2a

To calculate the results of one quantity by asking user to enter the rest of input values.

Source code

#include<iostream>

#include<conio.h>

#include<math.h>

using namespace std;

int acceleration(int vf, int vi, int s)

{ int acceleration=(vf\*vf-vi\*vi)/(2\*s);

return acceleration;

}

int Distance(int vf, int vi, int a)

{

int Distance=(vf\*vf-vi\*vi)/(2\*a);

return Distance;

}

int final\_vilocity(int vi, int s, int a)

{

int final\_velocity=sqrt(vi\*vi+2\*a\*s);

return final\_velocity;

}

int initial\_vilocity(int vf, int s, int a)

{

int initial\_velocity=sqrt(vf\*vf+2\*a\*s);

return initial\_velocity;

}

int main()

{ int x;

double vf1, vi1, s1, a1;

cout<<"write 1 to get accelarion\n2 to get distance\n3 to get final velocity\n4 to get initial velocity"<<endl;

cin>>x;

switch(x)

{

case 1:

cout<<"write final velocity"<<endl;

cin>>vf1;

cout<<" write initial velocity"<<endl;

cin>>vi1;

cout<<" write distance"<<endl;

cin>>s1;

cout<<acceleration(vf1, vi1, s1);

break;

case 2:

cout<<"write final velocity"<<endl;

cin>>vf1;

cout<<" write initial velocity"<<endl;

cin>>vi1;

cout<<" acceleration"<<endl;

cin>>a1;

cout<<Distance(vf1, vi1, a1);

break;

case 3:

cout<<"distance"<<endl;

cin>>s1;

cout<<" write initial velocity"<<endl;

cin>>vi1;

cout<<" acceleration"<<endl;

cin>>a1;

cout<<final\_vilocity(vi1, s1, a1);

break;

case 4:

cout<<"distance"<<endl;

cin>>s1;

cout<<" write final velocity"<<endl;

cin>>vf1;

cout<<" acceleration"<<endl;

cin>>a1;

cout<<initial\_vilocity(vf1, s1, a1);

break;

default:

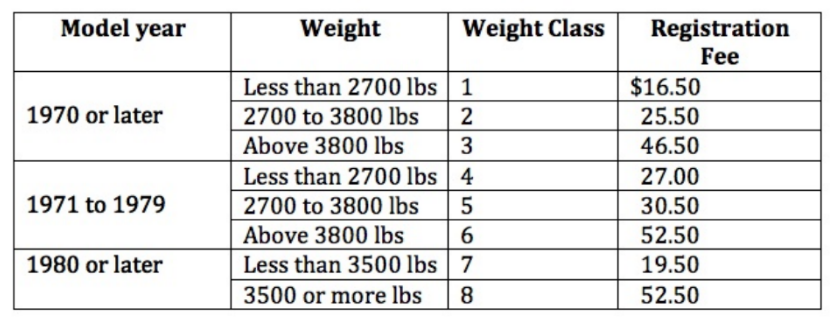
cout<<" wrong input ";

}

getch();

}

1. Based on an automobile’s model year and weight the state of New jersey determines the car’s weight class and registration fee using the following schedule:



Using this information, write a C++ function that accepts the year and weight in pounds(lbs) of an automobile and determines and displays the weight class and registration fee for a car.

Source code

#include<iostream>

#include<conio.h>

#include<math.h>

#include<iomanip>

using namespace std;

void company(int year, int weight){

if(year<=1970){

if(weight<2700){

cout<<"the wight class is 1\nthe price of the registration is $16.50"<<endl;}

else if(weight>=2700 && weight<=3800){

cout<<"the wight class is 2\nthe price of the registration is $25.50";}

else if(weight>3800){

cout<<"the wight class is 3\nthe price of the registration is $46.50";}

}

else if(year>=1971&&year<=1979){

if(weight<2700){

cout<<"the wight class is 4\nthe price of the registration is $27.00"<<endl;}

else if(weight>=2700 && weight<=3800){

cout<<"the wight class is 5\nthe price of the registration is $30.50";}

else if(weight>3800){

cout<<"the wight class is 6\nthe price of the registration is $52.50";}

}

else if(year>=1980){

if(weight<3500){

cout<<"the wight class is 7\nthe price of the registration is $19.50"<<endl;}

else if(weight>=3500){

cout<<"the wight class is 8\nthe price of the registration is $52.50";}

}

}

int main()

{int yr,wt;

cout<<" write year"<<endl;

cin>>yr;

cout<<" write weight"<<endl;

cin>>wt;

company(yr, wt);

getch();

return 0;

}

1. Write a function that separates each digit of a five digit integer and display them separately. For example, if user enter 87498, it should display  
   8 7 4 9 8.

You should design a function to separate digits.

Source code

#include<iostream>

#include<conio.h>

#include<math.h>

using namespace std;

int space(int a)

{int z;

for(int i=4;i>=0;i--)

{

z=pow(10,i);

cout<<a/z;

a=a%z;

cout<<" ";

}

}

int main()

{

int a;

cout<<"Enter 5 digit number "<<endl;

cin>>a;

space(a);

}

1. LCM is least common multiple of two numbers. It is the smallest number that divides both numbers. For example, LCM of 12 and 24 is 3. Although, both numbers are divisible by 6 and 12 as well, however, 3 is the smallest number that divides both. Design a function that takes two number as formal parameters, return LCM of both numbers.

Source code

#include<iostream>

#include<conio.h>

#include<math.h>

#include<iomanip>

using namespace std;

void lcm1(int a,int b){

int c=a\*b;

while(a!=b){

if(a>b)

a=a-b;

}

cout<<" lcm is of two number is "<<c/a;

}

int main()

{ int z,x;

cout<<" write two number and get the lcm "<<endl;

cin>>z>>x;

lcm1(z,x);

getch();

}

1. Write a program that takes as input your gross salary and your total saving and uses another function named taxCalculator() to calculate your tax. The taxCalculator() function takes as parameters the gross salary as well as the total savings amount. The tax is calculated as follows:  
   (a) The savings is deducted from the gross income to calculate the taxable income. Maximum deduction of savings can be Rs. 100,000, even though the amount can be more than this.  
   (b) For up to 100,000 as taxable income the tax is 0 (Slab 0); beyond 100,000 to 200,000 tax is 10% of the difference above 100,000 (Slab 1); beyond 200,000 up to 500,000 the net tax is the tax calculated from Slab 0 and Slab 1 and then 20% of the taxable income exceeding 200,000 (Slab 2); if its more than 500,000, then the tax is tax from Slab 0, Slab 1, Slab 2 and 30% of the amount exceeding 500,000.

Source code

#include<iostream>

#include<conio.h>

using namespace std;

int taxCalculator(int gs,int ts)

{

int max,slab1,slab2,slab3,slab4;

max=gs-ts;

if(max>0 && max<=100000)

{

slab1=max\*0.0;

cout<<"AMOUNT"<<slab1<<endl;

}

else if(max>100000 && max<=200000)

{

slab2=(max-10000)\*0.1+slab1;

cout<<"AMOUNT"<<slab2<<endl;

}

else if(max>200000 && max<=500000)

{

slab3=(max-200000)\*0.2+slab1+slab2;

cout<<"AMOUNT"<<slab3<<endl;

}

else if(max>500000)

slab4=(max-500000)\*0.3+slab1+slab2+slab3;

cout<<"AMOUNT"<<slab4<<endl;

}

int main()

{

int gs,ts;

cout<<"ENTER THE GROSS SALARY"<<endl;

cin>>gs;

cout<<"ENTER THE SAVING SALARY"<<endl;

cin>>ts;

taxCalculator(gs,ts);

}