DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

A

MINI PROJECT REPORT

ON

"CONVERSION SYSTEM USING C++"

Submitted in the partial fulfillment of the requirements in the 4th semester of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

BY

SHALINI R S

1NH17IS095

Under the guidance of

Mrs. Shwetha K S,

Sr. Assistant Professor,

Dept. of ISE, NHCE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING

(Autonomous College Permanently Affiliated to VTU, Approved by AICTE, Accredited by NBA & NAAC with 'A' Grade)

Ring Road, Bellandur Post, Near Marathalli,

Bangalore-560103, INDIA

CERTIFICATE

Certified that, the mini project report entitled **"CONVERSION SYSTEM USING C++"** carried out by SHALINI R S (1NH17IS095), a bonafied student of New Horizon College of Engineering, Bengaluru, in partial fulfillment of the requirements in the IV semester of Bachelor of Engineering in Information Science and Engineering the year 2018-2019. The project report has been approved as it satisfies the academic requirement in respect of mini project work.

Name & Signature of Guide

(Mrs. Shwetha K S)

Name & signature of HOD

(Dr. R J Anandhi)

Chapter 01

INTRODUCTION

The mini project `Conversion System` is an application developed for converting units from one form to a desired form. The basic operation of the Conversion System is simple: you enter the units that you want to convert from and the units that you want to convert to. You can use the program interactively with prompts. Sciences are built upon measurements. Measurements are expressed with numbers. This allows the logic, precision and power of mathematics to be brought to bear on our study of nature.

Units of measurement are names which characterize the kind of measurement and the standard of comparison to which each is related. So, when we see a measurement expressed as "7.5 feet" we immediately recognize it as a measurement of length, expressed in the unit "foot" (rather than other possible length units such as yard, mile, meter, etc.) Since many possible units are available for any measurement it is essential that every measurement include the unit name. A statement such as "the length is 7.5" is ambiguous, and therefore meaningless.

A single numeric conversion factor is associated with each simple unit. The conversion factor is the number by which a quantity expressed in that unit must be multiplied in order to be expressed in the equivalent unit in the standard system of units. We have chosen as our standard the SI system of units; a different standard system could be chosen without affecting the algorithms described here. Thus, the conversion factor for meter is 1.0, while the conversion factor for foot is 0.3048 since 1 foot = 0.3048 meter. The conversion factor for a numeric constant is just the constant itself.

This system is implemented to convert all the possible units of various physical quantities from one form to a desired form in a single step. This Conversion System uses basic C++ functions and concepts to generate menus, display messages, print texts on the screen and many more. The above proposed system implements the concept of classes and objects using C++ to define the various conversions along with various other concepts of C++ such as

switch statements, looping and branching constructs and strings and string manipulation functions.

1.1 Motivation of project

Units play a vital role in any calculations that are being performed. Without units all the calculations and conversions would be meaningless. Even in the digital era that we are living in, many calculations are done manually. The manual conversion of units has its own set of disadvantages which includes lack of accuracy, lack of reliability and consumption of more time.

To overcome the disadvantages discussed above the application `Conversion System` is developed. The `Conversion System` allows the user to perform various conversions on a single platform. This system displays a main menu of all the conversions that are possible from which the user can choose the desired option and perform the conversion by entering the values. This program proves to be effective as all the conversions can be performed on a single platform and produces accurate results.

1.2 Problem Statement

"To design and implement a Unit Conversion System in C++ to convert different kinds of units from one form to another."

Chapter 02

SYSTEM REUIREMENT SPECIFICATION

Purpose: The main purpose of the conversion system is that it reduces the work involved in converting the units manually. Through the above proposed system the units can be converted efficiently.

The purposes of developing of a conversion system are:

1. To implement different constructs of C++ language like lops, branching constructs, switch cases and many more.
2. To be able to solve different problem statements in C++.
3. To implement the concept of classes and objects in C++.
4. To understand different concept of C++.

The objectives of the conversion system are:

1. To eliminate the errors involved in conversion of units manually.
2. To simplify different processes of conversion of units and make it more efficient.
3. To reduce the paper work involved in the manual conversion of units.
4. To make the conversions more accurate and error free.
5. To save the user's time.

Scope:

1. Ensures effective conversion of units.
2. To be able to performs all conversions on a single platform.
3. Any number of units can be converted from one form to another easily.
4. The conversions of various physical quantities can be performed in a single step which makes the process of conversion much simpler.

2.1 Hardware System Configuration

Processor	-Intel Core i5
Speed	-1.8 GHz
RAM	-256 MB (min)
Hard disk	-10 GB

2.2 Software System Specifications

Operating System	-Windows 8
Programming	-C++ language
Language Compiler	-Code:: blocks

Chapter 03

METHODOLOGY

3.1 Algorithm

MAINMENU FUNCTION

The above proposed conversion system enables the user to perform conversions of various physical quantities such as length, temperature, weight, area and volume. Using this system the user can convert one system of units to another.

A main menu is displayed on the screen which displays the different conversions that can be performed in the application. Based on the user's choice the particular class will be called and the task can be performed. The algorithm for the above function is as follows-

STEPS:

I. START

II. Declare objects for all the classes.

III. While(1)

The mainmenu is displayed as follows:

1. LENGTH
2. TEMPERATURE
3. WEIGHT
4. AREA
5. VOLUME
6. EXIT

IV. The user is asked to enter the choice.

V. Based on the choice of the user the particular class is called.

VI. If(x==1)

class length is called and the menu is displayed as below:

1 : mm-m

2 : m-mm

3 : cm-m

4 : m-cm

5 : cm_km

6 : km-cm

7 : m-mile

8 : mile-m

9 : km-mile

10 : mile-km

11 : feet-m

12 : m-feet

13 : inch-m

14 : m-inch

15 : yard-m

16 m-yard

17 : Back to The Main Menu

cout<<"Enter the choice";

cin>>y;

Based on the value of 'y', the particular conversions are performed.

VII. If(x==2)

class temperature is called and the menu is displayed as below:

1 : Celsius-farenheit

2 : farenheit-celsius

3 : Celsius-kelvin

4 : kelvin-celsius

5 : Back to The Main Menu

cout<<"Enter the choice";

cin>>y;

Based on the value of 'y', the particular conversions are performed.

VIII. if(x==3)

class weight is called and the menu is displayed as below:

1: Milligm-Gramm

2: Gramm-milligm

3: Gramm-killogram

4: killoGramm-Gramm

5: pound-killogramm

6: killogramm-pound

7: Gramm-Pound

8: Pound-gramm

9: killogramm-Metric ton

10: Metric ton-Killogramm

11: Back to The Main Menu

cout<<"Enter the choice";

cin>>y;

Based on the value of 'y', the particular conversions are performed.

IX. if(x==4)

class weight is called and the menu is displayed as below:

1: Square mm-Square cm

2: square cm-Square mm

3: square cm-square m

4: Square m-Square cm

5: Square m-Square km

6: Square km-Square m

7: Square feet-Square m

8: Square m-Square feet

9: Square Yard-Square m

10: Square m-Square yard

11: Square mile-Square km

12: Square km-Square mile

13: Acre-Hectare

14: Hectare-Acre

15: Square km-Acre

16: Acre-Square km

17: Back to The Main Menu

cout<<"Enter the choice";

cin>>y;

Based on the value of 'y', the particular conversions are performed.

X. if(x==5)

class weight is called and the menu is displayed as below:

1 : ml-Litre";

2 : Litre-ml";

3 : Cubic mm-Cubic cm

4 : Cubic cm-Cubic mm

5 : Cubic cm-Cubic m

6 : Cubic m-Cubic cm

7 : Cubic Inch-Cubic m

8 : Cubic m-Cubic Inch

9 : Cubic feet-Cubic m

10 : Cubic m-Cubic feet

11 : Cubic m-Gallon (uk)

12 : Gallon-Cubic m

13 : Litre-Gallon

14 : Gallon-Litre

15 : Back to The Main Menu

cout<<"Enter the choice";

cin>>y;

Based on the value of 'y', the particular conversions are performed.

XI. END

3.3 Code and implementation

```
#include<iostream>
#include<conio.h>
using namespace std;
class input
{
  public:
        float i;

};
  class length:public input
  {
   public:
       void mm_m();
       void m_mm();
       void cm_m();
       void m_cm();
       void cm_km();
       void km_cm();
       void m_mile();
       void mile_m();
       void km_mile();
       void mile_km();
       void feet_m();
       void m_feet();
       void inch_m();
       void m_inch();
       void yard_m();
       void m_yard();
```

```
    };
class temp:public input
 {
 public:
      void cel_f();
      void f_cel();
      void cel_k();
      void k_cel();
      void f_k();
      void k_f();

  };

  class weight:public input
    {
    public:
      void milligm_gm();
      void gm_milligm();
      void gm_kg();
      void kg_gm();
      void kg_mton();
      void mton_kg();
      void pound_kg();
      void kg_pound();
      void gm_pound();
      void pound_gm();

    };

   class area :public input
```

```
{
public:
   void mmSq_cmSq();
   void cmSq_mmSq();
   void cmSq_mSq ();
   void mSq_cmSq ();
   void mSq_kmSq ();
   void kmSq_mSq ();
   void feetSq_mSq();
   void mSq_feetSq();
   void yardSq_mSq();
   void mSq_yardSq();
   void mileSq_kmSq();
   void kmSq_mileSq();
   void acre_kmSq();
   void kmSq_acre();
   void acre_hect();
   void hect_acre();
    };

class vol:public input
{
public:
   void mL_L();
   void L_mL();
   void mmQ_cmQ();
   void cmQ_mmQ();
   void cmQ_mQ ();
   void mQ_cmQ ();
   void inchQ_mQ();
```

```
    void mQ_inchQ();
    void feetQ_mQ();
    void mQ_feetQ();
    void mQ_gallon();
    void gallon_mQ();
    void L_gallon();
    void gallon_L();

};

 void length ::mm_m()
    {
     cout<<"\n   Millimeter= ";
     cin>>i;
     cout<<"\n   Meter= "<<i*.001;
     }
 void length ::m_mm()
    {
    cout<<"\n   meter= ";
    cin>>i;
    cout<<"\n   millimeter= "<<i*1000;
    }
 void length ::cm_m()
    {
     cout<<"\n   Centimeter= ";
     cin>>i;
     cout<<"\n   Meter= "<<i*.01;
     }
 void length ::m_cm()
    {
```

```
    cout<<"\n   meter= ";
    cin>>i;
    cout<<"\n   cm= "<<i*100;
    }
void length::m_mile()
    {
    cout<<"\n   meter= ";
    cin>>i;
    cout<<"\n   mile= "<<i*.00062;
    }
void length::mile_m()
   {
   cout<<"\n   mile= ";
   cin>>i;
   cout<<"\n   meter= "<<i*1609.344;
   }
 void length :: cm_km()
   {
    cout<<"\n   cm= ";
    cin>>i;
    cout<<"\n   km= "<<i*0.00001;

   }

void length :: km_cm()
   {   cout<<"\n   km= ";
      cin>>i;
      cout<<"\n   cm= "<<i*100000;
   }
void length::feet_m()
```

```
    {
    cout<<"\n   feet= ";
    cin>>i;
    cout<<"\n   meter= "<<i*.3048;
    }
  void length::m_feet()
    {
    cout<<"\n   meter= ";
    cin>>i;
    cout<<"\n   feet= "<<i*3.28084;
    }
  void length::yard_m()
    {
    cout<<"\n   yard= ";
    cin>>i;
    cout<<"\n   meter= " <<i*0.9144;
    }
  void length::m_yard()
    {
    cout<<"\n   meter= ";
    cin>>i;
    cout<<"\n   yard= "<<i*1.09361;
    }
  void length::inch_m()
    {
    cout<<"\n   inch=";
    cin>>i;
    cout<<"\n   meter="<<i*.0254;
    }
  void length::m_inch()
```

```
    {
    cout<<"\n   meter=";
    cin>>i;
    cout<<"\n   inch="<<i*39.37008;
    }
void length ::km_mile()
    {
    cout<<"\n   kilometer=";cin>>i;
    cout<<"\n   mile="<<i*.6213712 ;
    }
void length ::mile_km( )
    {
    cout<<"\n   mile="; cin>>i;
    cout<<"\n   kilometer="<<i*1.60934;
    }



void temp::cel_f()
   {
   cout<<"\n   Celsius= ";
   cin>>i;
    cout<<"\n   Fahrenheit= " <<(((9*i)/5)+32);
   }
void temp::f_cel()
   {
   cout<<"\n   Fahrenheit= ";   cin>>i;
   cout<<"\n   Celsius= "<<(((i-32)/9)*5);
    }
void temp::cel_k()
```

```
    {
    cout<<"\n   celsius= ";cin>>i;
    cout<<"\n   kelvin= "<<i+273;
    }
void temp::k_cel()
    {
    cout<<"\n   kelvin= ";  cin>>i;
    cout<<"\n   celsius= "<<i-273;
    }



void weight::milligm_gm()
    {
    cout<<"\n   milligramm= ";cin>>i;
    cout<<"\n   gramm= "<<i*.001;
    }
void weight::gm_milligm()
    {
    cout<<"\n   gram= "; cin>>i;
    cout<<"\n   milligramm= "<<i*1000;
    }
void weight::pound_kg()
   {
   cout<<"\n   pound= "; cin>>i;
   cout<<"\n   kilogramm= "<<i*.45359;
   }
 void weight::gm_kg()
    { cout<<"\n   Gramm= ";cin>>i;
     cout<<"\n   killogramm= "<<i*.001;
```

```
    }
 void weight::kg_gm()
{    cout<<"\n   killogramm= ";cin>>i;
    cout<<"\n   Gramm= "<<i*1000;
    }


void weight::kg_pound()
 {
 cout<<"\n   kilogramm= ";cin>>i;
 cout<<"\n   pound= "<<2.20462*i;
 }
void weight::gm_pound()
 {
 cout<<"\n   gramm= ";  cin>>i;
 cout<<"\n   pound= "<<i*.0022;
 }
void weight::pound_gm()
 {
 cout<<"\n   pound= "; cin>>i;
 cout<<"\n   gramm= "<<i*453.59237;
 }
void weight::kg_mton()
 {
 cout<<"\n   kilogramm= ";cin>>i;
 cout<<"\n   matric ton= "<<i*.001;
  }
void weight::mton_kg()
 {
 cout<<"\n   metric ton= "; cin>>i;
```

```
 cout<<"\n   kilogramm= "<<i*1000;
  }


void area::mmSq_cmSq()
   {
   cout<<"\n   squqre mm= ";cin>>i;
   cout<<"\n   square cm= "<<i*.01;
   }
void area::cmSq_mmSq()
   {
   cout<<"\n   square cm= ";cin>>i;
   cout<<"\n   square mm= "<<i*100;
   }
void area::cmSq_mSq()
   {
   cout<<"\n   square cm= ";cin>>i;
   cout<<"\n   square m= "<<i*.0001;
   }
void area::mSq_cmSq()
   {
   cout<<"\n   square m= "; cin>>i;
   cout<<"\n   square cm= "<<i*10000;
   }
void area::mSq_kmSq()
   {
   cout<<"\n   square m= "; cin>>i;
   cout<<"\n   square km= "<<i*.000001;
   }
void area::kmSq_mSq()
```

```
    {
    cout<<"\n   square km= ";cin>>i;
    cout<<"\n   square m= "<<i*1000000;
    }
void area::feetSq_mSq()
 {
 cout<<"\n   square feet= ";cin>>i;
 cout<<"\n   square m= "<<i*.0929;
 }
void area::mSq_feetSq()
 {
 cout<<"\n   square m= ";  cin>>i;
 cout<<"\n   square feet= "<<i*10.76391;
 }
void area::yardSq_mSq()
 {
 cout<<"\n   square yard= ";cin>>i;
 cout<<"\n   square m= "<<i*.83613;
 }
 void area::mSq_yardSq()
  {
   cout<<"\n   square m= "; cin>>i;
   cout<<"\n   square yard= "<<i*1.19599;
  }
 void area::mileSq_kmSq()
   {
   cout<<"\n   square mile= ";cin>>i;
   cout<<"\n   square km= "<<i*2.5899;
   }
 void area::kmSq_mileSq()
```

```
  {
   cout<<"\n   square km= " ;cin>>i;
    cout<<"\n   square mile= "<<i*.3861;
  }
void area::acre_hect()
  {
  cout<<"\n   Acre= ";   cin>>i;
  cout<<"\n   Hector= "<<i*.40469;
  }
void area::hect_acre()
  {
  cout<<"\n   Hector= ";cin>>i;
  cout<<"\n   Acre= " <<i*2.47105;
  }
 void area::acre_kmSq()
   {cout<<"\n   Acre= ";cin>>i;
   cout<<"\n   Square km= "<<i*.00405;
   }
 void area::kmSq_acre()
 {
  cout<<"\n   Square km= ";cin>>i;
  cout<<"\n   Acre= "<<i*247.10538;

 }

void vol::mL_L()
   {
   cout<<"\n   milli litre= ";cin>>i;
   cout<<"\n   Litre= "<<i*.001;
   }
```

```
void vol::L_mL()
  {       cout<<"\n   Litre= "; cin>>i;
  cout<<"\n   milli Litre= "<<i*1000;
  }
void vol::cmQ_mmQ()
  {
   cout<< "\n   cubic cm= "; cin>>i;
   cout<<"\n   cubic mm= "<<i*1000;
   }
void vol::mmQ_cmQ()
  {
  cout<<"\n   cubic mm= ";   cin>>i;
  cout<<"\n   cubic cm= "<<i*.001;
  }
void vol::cmQ_mQ()
  {
  cout<<"\n   cubic cm= ";  cin>>i;
  cout<<"\n   cubic meter= "<<i*.000001;
  }
void vol::mQ_cmQ()
  {
  cout<<"\n   cubic meter= "; cin>>i;
  cout<<"\n   cubic cm= "<<i*1000000;
  }
void vol::inchQ_mQ()
 {
 cout<<"\n   cubic inch= ";  cin>>i;
 cout<<"\n   cubic meter= "<<i*.00002;
 }
void vol::mQ_inchQ()
```

```
 {
 cout<<"\n   cubic meter= "; cin>>i;
 cout<<"\n   cubic inch= "<<i*61023.74409;
 }
void vol::feetQ_mQ()
 {
 cout<<"\n   cubic feet= "; cin>>i;
 cout<<"\n   cubic meter= "<<i*1222 ;

 }
void vol::mQ_feetQ()
 {
 cout<<"\n   cubic meter= ";cin>>i;
 cout<<"\n   cubic feet= "<<i*123;
 }
void vol::mQ_gallon()
 {
 cout<<"\n   cubic meter= ";cin>>i;
 cout<<"\n   gallon= "<<i*264.17205;
 }
void vol::gallon_mQ()
 {
 cout<<"\n   Gallon= "; cin>>i;
 cout<<"\n   cubic meter= "<<i*.00379;
 }
void vol::L_gallon()
  {
  cout<<"\n   Litre= ";cin>>i;
  cout<<"\n   gallon= "<<i*.21997;
  }
```

```
void vol::gallon_L()
   {
   cout<<"\n   Gallon= "; cin>>i;
   cout<<"\n   Litre= "<<i*4.54609;
   }



int main()
   {
length b;
weight c;
vol d;
area e;
temp f;
int x,y;

while(1)
{
   cout<<"\t\t\tWELCOME TO UNIT CONVERTION\n";
   cout<<"\n      TYPE"<<"\n     ------";
   cout<<"\n   1: Length\n" << "   2: Temparature\n"<<"   3: Weight\n"<<"   4: Area\n"<<"
5: Volume\n" <<"  6: Exit\n";
   cout<<"\n\nPlease choose your Convertion Type:" ;
   cin>>x;
   if(x==1)
      {

         cout<<"\n\n choose your unit convertion:\n";
```

```
        cout<<"\n  1 : mm-m";
        cout<<"\n  2 : m-mm";
        cout<<"\n  3 : cm-m";
        cout<<"\n  4 : m-cm";
        cout<<"\n  5 : cm_km";
        cout<<"\n  6 : km-cm";
        cout<<"\n  7 : m-mile";
        cout<<"\n  8 : mile-m";
        cout<<"\n  9 : km-mile";
       cout<<"\n 10 : mile-km";
       cout<<"\n 11 : feet-m";
       cout<<"\n 12 : m-feet";
       cout<<"\n 13 : inch-m";
       cout<<"\n 14 : m-inch";
       cout<<"\n 15 : yard-m";
      cout<<"\n 16 : m-yard";
      cout<<"\n 17 : Back to The Main Menu";
while(1)
    {
     cout<<"\n\n Please Enter Your Choice= ";
    cin>>y;

     if (y==1)
        { b.mm_m(); }
     else if(y==2)
        { b.m_mm(); }
     else if (y==3)
        { b.cm_m(); }
     else if (y==4)
        { b.m_cm(); }
```

```
    else if (y==5)
       { b.cm_km(); }
    else if (y==6)
       {  b.km_cm(); }
    else if (y==7)
       {  b.m_mile(); }
    else if (y==8)
       {   b.mile_m(); }
    else if (y==9)
        {   b.km_mile(); }
    else if (y==10)
       {  b.mile_km(); }
    else if (y==11)
        {  b.feet_m(); }
    else if (y==12)
        { b.m_feet(); }
     else if (y==13)
         { b.inch_m();}
    else if(y==14)
       { b.m_inch();}
    else if (y==15)
       {b.yard_m();}
  else if (y==16)
       {b.m_yard();}
  else if (y==17)
       {break;}
}
}
else if(x==2)
    {
```

```
        cout<<"\n\n choose your unit convertion:\n";

        cout<<"\n  1: Celsius-Fahrenheit";
        cout<<"\n  2: Fahrenheit-Celsius";
        cout<<"\n  3: Celsius-Kelvin";
        cout<<"\n  4: Kelvin-Celcius";
        cout<<"\n  5: Back to The Main Menu";
while(1)
{
cout<<"\n\n Please Enter Your Choice= ";
  cin>>y;

     if(y==1)
       { f.cel_f(); }
     else if(y==2)
       {f.f_cel();}
      else if (y==3)
       {f.cel_k();}
     else if(y==4)
      {f.k_cel();}
     else if(y==5)
        {break;}
  }
  }
  else if(x==3)
  {
   cout<<"\n\nchoose your unit convertion :\n";

   cout<<"\n  1: Milligm-Gramm";
   cout<<"\n  2: Gramm-milligm";
```

```
 cout<<"\n  3: Gramm-killogram";
 cout<<"\n  4: killoGramm-Gramm";
 cout<<"\n  5: pound-killogramm";
 cout<<"\n  6: killogramm-pound";
 cout<<"\n  7: Gramm-Pound";
 cout<<"\n  8: Pound-gramm";
 cout<<"\n  9: killogramm-Metric ton";
cout<<"\n 10: Metric ton-Killogramm";
cout<<"\n 11: Back to The Main Menu";

while(1)
 {
  cout<<"\n\nPlease Enter Your Choice= ";
  cin>>y;

 if (y==1)
    {c.milligm_gm();}
 else if (y==2)
    {c.gm_milligm();}
 else if (y==3)
    {c.gm_kg();}
 else if (y==4)
    {c.kg_gm();}
 else if (y==5)
    {c.pound_kg();}
 else if (y==6)
    {c.kg_pound();}
 else if (y==7)
    {c.gm_pound();}
 else if (y==8)
```

```
        {c.pound_gm();}
      else if (y==9)
        {c.kg_mton();}
      else if (y==10)
        {c.mton_kg();}
      else if (y==11)
        {break;}
  }
  }
  else if(x==4)
        {
        cout<<"\n\nchoose your unit convertion:\n";

        cout<<"\n   1: Square mm-Square cm";
        cout<<"\n   2: square cm-Square mm";
        cout<<"\n   3: square cm-square m";
        cout<<"\n   4: Square m-Square cm";
        cout<<"\n   5: Square m-Square km";
        cout<<"\n   6: Square km-Square m";
        cout<<"\n   7: Square feet-Square m";
        cout<<"\n   8: Square m-Square feet";
        cout<<"\n   9: Square Yard-Square m";
        cout<<"\n  10: Square m-Square yard";
        cout<<"\n  11: Square mile-Square km";
        cout<<"\n  12: Square km-Square mile";
        cout<<"\n  13: Acre-Hectare";
        cout<<"\n  14: Hectare-Acre";
        cout<<"\n  15: Square km-Acre ";
        cout<<"\n  16: Acre-Square km";
        cout<<"\n  17: Back to The Main Menu";
```

```
while(1)
{
cout<<"\n\nPlease Enter Your Choice= ";
  cin>>y;

    if(y==1)
      {e.mmSq_cmSq();}
    else if(y==2)
    {e.cmSq_mmSq();}
    else if(y==3)
    {e.cmSq_mSq();}
    else if(y==4)
    {e.mSq_cmSq();}
    else if(y==5)
    {e.mSq_kmSq();}
    else if(y==6)
    {e.kmSq_mSq();}
    else if(y==7)
    {e.feetSq_mSq();}
    else if(y==8)
    {e.mSq_feetSq();}
    else if(y==9)
    {e.yardSq_mSq();}
    else if(y==10)
    {e.mSq_yardSq();}
    else if(y==11)
    {e.mileSq_kmSq();}
    else if(y==12)
    {e.kmSq_mileSq();}
```

```
        else if(y==13)
        {e.acre_hect();}
        else if(y==14)
        {e.hect_acre();}
        else if(y==15)
        {e.kmSq_acre();}
        else if(y==16)
        {e.acre_kmSq();}
        else if(y==17)
        {break;}
    }
    }
else if(x==5)
    {
      cout<<"\n\nchoose your unit convertion:\n";

      cout<<"\n   1 : ml-Litre";
      cout<<"\n   2 : Litre-ml";
      cout<<"\n   3 : Cubic mm-Cubic cm";
      cout<<"\n   4 : Cubic cm-Cubic mm";
      cout<<"\n   5 : Cubic cm-Cubic m";
      cout<<"\n   6 : Cubic m-Cubic cm";
      cout<<"\n   7 : Cubic Inch-Cubic m";
      cout<<"\n   8 : Cubic m-Cubic Inch";
      cout<<"\n   9 : Cubic feet-Cubic m";
      cout<<"\n  10 : Cubic m-Cubic feet";
      cout<<"\n  11 : Cubic m-Gallon (uk)";
      cout<<"\n  12 : Gallon-Cubic m";
      cout<<"\n  13 : Litre-Gallon";
      cout<<"\n  14 : Gallon-Litre";
```

```
   cout<<"\n  15 : Back to The Main Menu";

while(1)
 { cout<<"\n\nPlease Enter Your Choice= ";
  cin>>y;

    if (y==1)
      {d.mL_L();}
 else if (y==2)
      {d.L_mL();}
 else if (y==3)
      {d.mmQ_cmQ();}
else if (y==4)
      {d.cmQ_mmQ();}
else if (y==5)
      {d.cmQ_mQ();}
else if (y==6)
      {d.mQ_cmQ();}
else if (y==7)
      {d.inchQ_mQ();}
else if (y==8)
      {d.mQ_inchQ();}
else if (y==9)
      {d.feetQ_mQ();}
else if (y==10)
      {d.mQ_feetQ();}
else if (y==11)
      {d.mQ_gallon();}
else if (y==12)
      {d.gallon_mQ();}
```

```
        else if (y==13)
            {d.L_gallon();}
        else if(y==14)
            {d.gallon_L();}
        else if(y==15)
            {break;}
       }
       }
    else if(x==6){ break;}
      }
}
```

Chapter 04

RESULTS AND DISCUSSION

Fig 4.1 main menu displayed

The main menu of all the physical quantities like length, temperature, weight, area and volume is shown and the user is asked to enter his conversion type.

Fig 4.2 length conversion menu displayed and conversion being performed

As the user has entered the choice as one, all possible length conversions are displayed and the user is asked to enter the choice. Once the user enters his choice, the values are asked and the conversion is being performed.

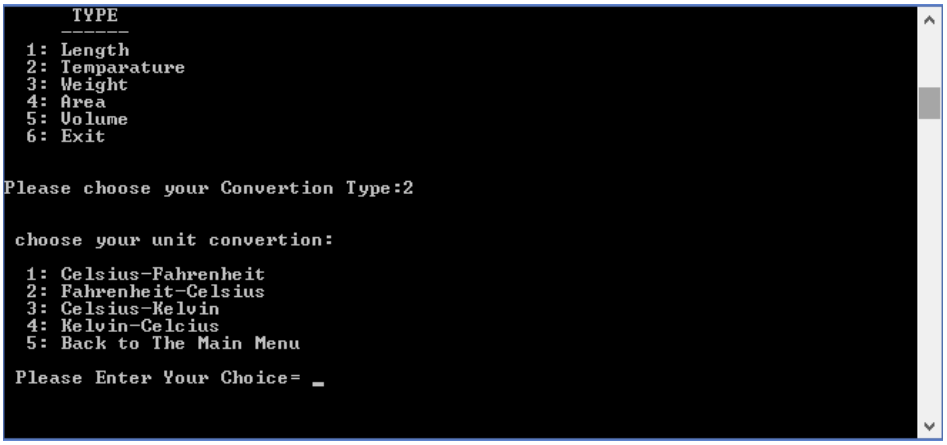

Fig 4.3 temperature conversion menu displayed

As the user has entered the choice as 2, all the possible temperature conversions are displayed and the user is to enter the choice.

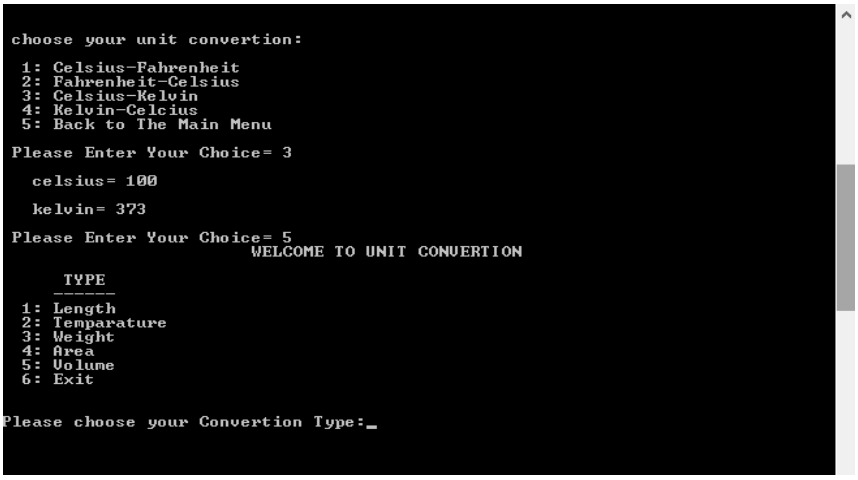

Fig 4.4 temperature conversion being performed and main menu displayed.

After entering the choice, the user is asked to enter the values and the conversion is performed and the results are displayed.

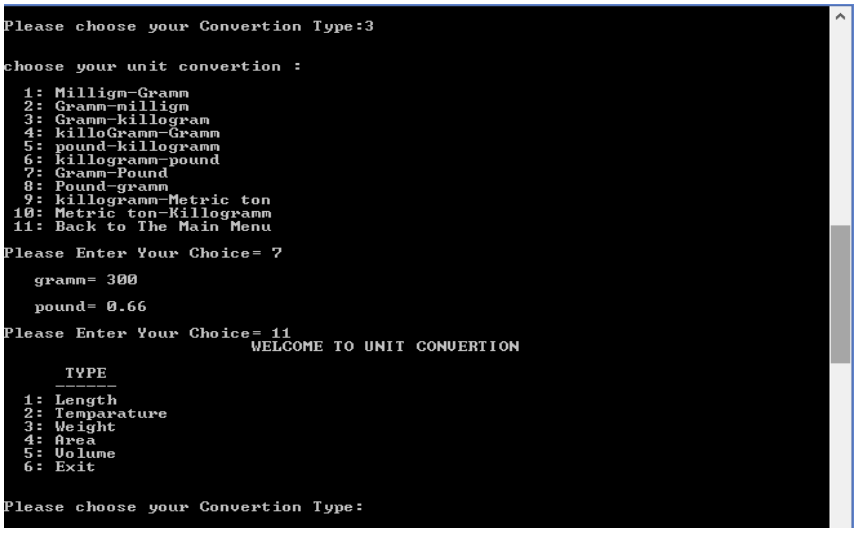

Fig 4.5 weight conversion menu displayed and conversion being performed.

As the user has entered the choice as 3, all possible weight conversions are displayed and the user is asked to enter the choice. After entering the choice, the user is asked to enter the values and the conversion is performed

.

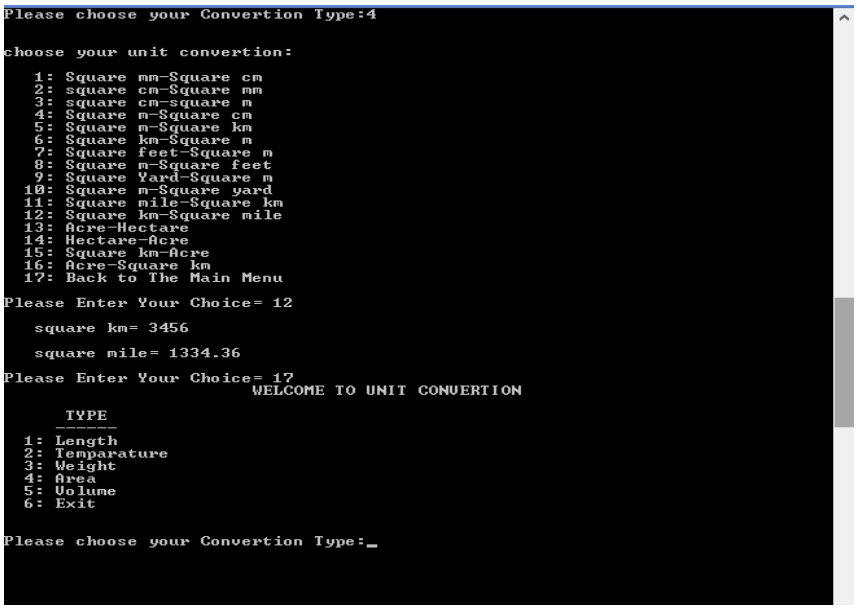

Fig 4.6 area conversion menu displayed and conversion being performed.

As the user has entered the choice as 4, all possible area conversions are displayed and the user is asked to enter the choice. After entering the choice, the user is asked to enter the values and the conversion is performed.

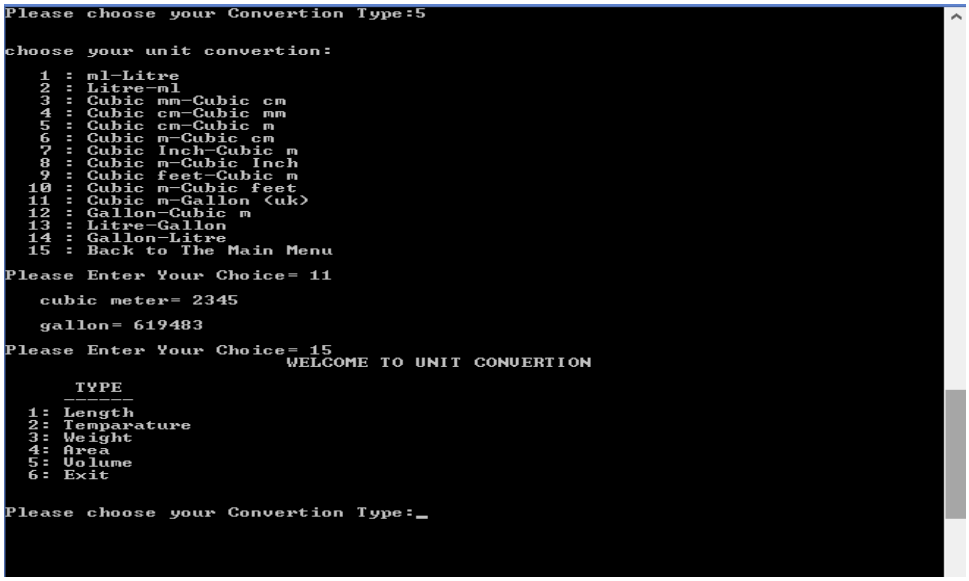

Fig 4.7 volume conversion menu displayed and conversion being performed.

As the user has entered the choice as 5, all possible volume conversions are displayed and the user is asked to enter the choice. After entering the choice, the user is asked to enter the values and the conversion is performed.

CONCLUSION AND FUTURE ENHANCEMENTS

As conversion of units manually has its own set of disadvantages including lower efficiency, higher time consumption and high risk of errors, the above mentioned disadvantages can be eliminated by using the above proposed conversion system. The unit conversion system proves to be an user friendly application and has various advantages over the manual conversion of units which includes quick, easy and error free conversions. This application allows the user to perform all complex conversions in a simpler way on a single platform. This method is much more user-friendly and time efficient and therefore it would be much more easier and convenient for the users.

REFERENCES

[1] https://www.cs.utexas.edu/users/novak/units95.html

[2] https://www.gnu.org/software/units/manual/units.pdf

[3] https://www.nist.gov/pml/weights-and-measures/metric-si/unit-conversion

[4] http://cppprojectsolve.blogspot.com/2010/09/si-no_05.html?m=1

[5] http://www.cplusplus.com/forum/beginner/139635/

[6] http://cppprojectcode.blogspot.com/2010/09/unit-conversion-software.html

[7] https://www.cengage.com/resource_uploads/downloads/1111136025_277026.pdf

[8] https://en.wikipedia.org/wiki/Conversion_of_units