CS211 OBJECT ORIENTED PROGRAMMING

Submitted To**: **Ms. SHAISTA RASHID

Submitted by:

ZAINAB TANVEER	***3684FBAS/BSSE/F18A***
QURATULAIN	***3689/FBAS/BSSE/F18A***
HAFSA HASSAN ABBASI	***3641/FBAS/BSSE/F18A***

SUBMISSION ON:

20-02-2019

QUESTION 01:

```
#pragma once
class date
{
private:
       int day;
       int month;
       int year;
public:
       date();
       date(int d, int m, int y);
       bool valid_date();
       void get_date();
       void show_date();
       bool operator > (date d);
       bool operator < (date d);
       bool operator <= (date d);
       bool operator == (date d);
       bool operator != (date d);
       void operator - (date d);
       date operator --();
       date operator ++(int);
};
#include "stdafx.h"
#include "date.h"
#include <iomanip>
#include <iostream>
using namespace std;

date::date()
{
       day = 7;
       month = 1;
       year = 2000;
}
date::date(int d, int m, int y)
{
       day = d;
       month = m;
       year = y;
}
bool date::valid_date()
{
       if (month >= 1 && month <= 12 && day >= 1 && day <= 31 && year >= 1880 && year <=
2019)
       {
              if (month == 2)
              {
                     if (year % 4 == 0)
                     {
                            if (day <= 29)
                       {return true;}
else { return false; }
```

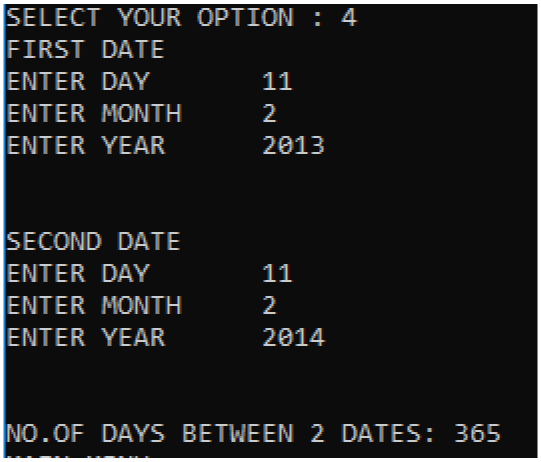
```
                }
                else if (year % 4 != 0)
                {
                    if (day <= 28)
                    {
                        return true;
                    }
                    else { return false; }
                }
            }

            if (month <= 7 && month % 2 == 0 && month != 2)
            {
                if (day <= 30)
                {
                    return true;
                }
                else { return false; }
            }
            else if (month <= 7 && month % 2 == 1)
            {
                if (day <= 31)
                {
                    return true;
                }
                else { return false; }
            }

            else if (month >= 8 && month % 2 == 0)
            {
                if (day <= 31)
                {
                    return true;
                }
                else { return false; }
            }
            else if (month >= 8 && month % 2 == 1)
            {
                if (day <= 30)
                {
                    return true;
                }
                else { return false; }
            }
        }
        else { return false; }
    }
    void date::get_date()
    {
        cout << "ENTER DAY\t";
        cin >> day;
        cout << "ENTER MONTH\t";
        cin >> month;
        cout << "ENTER YEAR\t";
        cin >> year;
        cout << endl << endl;
    }
    void date::show_date()
```

```
{
        cout << "DATE : ";
        cout << setw(2) << setfill('0') << day;
        cout << "/" << setw(2) << setfill('0') << month;
        cout << "/" << setw(4) << setfill('0') << year;
        cout << endl << endl << endl;
}
bool date::operator > (date d)
{
        if (year > d.year)
        {
                return true;
        }
        else if (year == d.year)
        {
                if (month > d.month)
                {
                        return true;
                }
                else if (month == d.month)
                {
                        if (day > d.day)
                        {
                                return true;
                        }
                        else return false;
                }
                else return false;
        }
        else return false;
}
bool date::operator < (date d)
{
        if (year < d.year)
        {
                return true;
        }
        else if (year == d.year)
        {
                if (month < d.month)
                {
                        return true;
                }
                else if (month == d.month)
                {
                        if (day < d.day)
                        {
                                return true;
                        }
                        else return false;
                }
                else return false;
        }
        else return false;
}
bool date::operator <= (date d)
{
        if (year < d.year)
```

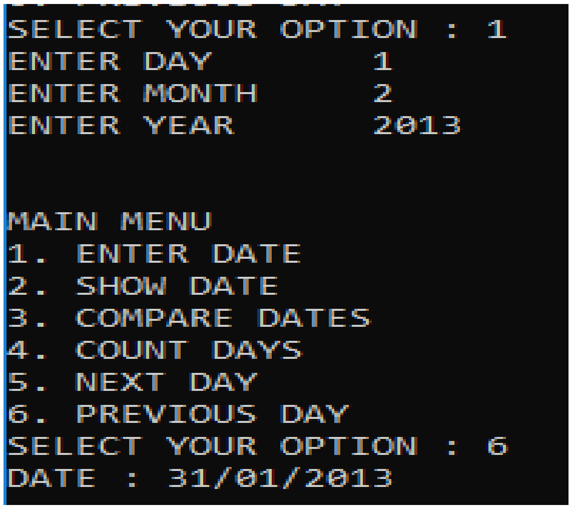
```
    {
        return true;
    }
    else if (year == d.year)
    {
        if (month < d.month)
        {
            return true;
        }
        else if (month == d.month)
        {
            if (day < d.day || day == d.day)
            {
                return true;
            }
            else
            {
                return false;
            }
        }
        else
        { return false; }
    }
    else
    { return false; }
}
bool date::operator == (date d)
{
    if (year == d.year)
    {
        if (month == d.month)
        {
            if (day == d.day)
            {
                return true;
            }
            else return false;
        }
        else return false;
    }
    else return false;
}
bool date::operator != (date d)
{
    if (year != d.year)
    {
        return true;
    }
    else if (year == d.year)
    {
        if (month != d.month)
        {
            return true;
        }
        else if (month == d.month)
{  if (day != d.day)
            {
                return true;
```

```
                        }
                        else return false;
                }
                else return false;
        }
        else return false;
}
void date::operator-(date d)
{

        int d1 = 0; int d2 = 0; int d3 = 0;
                        d1 = (year * 365) + (month * 31) + (day);
                        d2 = (d.year * 365) + (d.month * 31) + (d.day);

        if (d1>d2)
        {
                d3 = d1 - d2;
        }
        else
        {
                d3 = d2 - d1;
        }
        cout << "NO.OF DAYS BETWEEN 2 DATES: " << d3 << endl;

}
date date::operator++(int)
{
        date d1;
        d1.year = year;
        d1.month = month;
        d1.day = day;
        day = day + 1;
        if (valid_date() == false)
        {
                day = 1;
                month = month + 1;
        }
        if (month > 12)
        {
                day = 1;
                month = 1;
                year = year + 1;
        }

        return d1;
}
date date::operator--()
{
        date d1;

        day = day - 1;
        if (valid_date() == false)
        {
                if (month == 1)
                {
                        month = 12;
                        day = 31;
                        year = year - 1;
                }
                else
                {
                        day = 31;
                        month = month - 1;
                }
                if (valid_date() == false)
                {
                        if (month == 2 && year % 4 == 0)
                        {
                                day = 29;
                        }
                        else if (month == 2 && year % 4 != 0)
                        {
                                day = 28;
                        }
                        else
                        {
                                day = 30;
                        }
                }
                d1.year = year;
                d1.month = month;
                d1.day = day;
                return d1;
        }
}

// ConsoleApplication160.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "date.h"
#include <iostream>
using namespace std;

void main()
{
        date d1, d2, d3;
        int opt;
        do
        {
                cout << "MAIN MENU" << endl;
                cout << "1. ENTER DATE " << endl;
                cout << "2. SHOW DATE " << endl;
                cout << "3. COMPARE DATES  " << endl;
                cout << "4. COUNT DAYS " << endl;
                cout << "5. NEXT DAY" << endl;
                cout << "6. PREVIOUS DAY" << endl;
                cout << "SELECT YOUR OPTION : ";
                cin >> opt;
                if (opt == 1)
                {
                        d1.get_date();
                        if (d1.valid_date() == false)
                        {
                                cout << "INVALID" << endl;
                                cout << "YOUR DATE IS NOW SET TO 07/01/2000" << endl;
                d1 = date(07, 01, 2000);
                d1.show_date();
            }
        }
        if (opt == 2)
        {
            d1.show_date();
        }
        if (opt == 3)
        {
            cout << "FIRST DATE" << endl;
            d1.get_date();
            if (d1.valid_date() == false)
            {
                cout << "INVALID" << endl;
                cout << "YOUR DATE IS NOW SET TO 07/01/2000" << endl;
                d1 = date(07, 01, 2000);
                d1.show_date();
                cout << endl << endl;
            }
            cout << "SECOND DATE" << endl;
            d2.get_date();
            if (d2.valid_date() == false)
            {
                cout << "INVALID" << endl;
                cout << "YOUR DATE IS NOW SET TO 07/01/2000" << endl;
                d2 = date(05, 02, 2000);
                d2.show_date();
                cout << endl << endl;
            }
            cout << endl;
            if (d1 > d2)
            {
                cout << "-FIRST DATE IS GREATER THAN SECOND DATE" << endl;
            }
            if (d1 < d2)
            {
                cout << "-SECOND DATE IS GREATER THAN FIRST DATE" << endl;
            }
            if (d1 <= d2)
            {
                cout << "-FIRST DATE IS GREATER OR EQUAL THAN SECOND DATE" <<
endl;
            }
            if (d1 == d2)
            {
                cout << "-FIRST DATE IS EQUAL TO SECOND DATE" << endl;
            }
            if (d1 != d2)
            {
                cout << "-FIRST DATE IS NOT EQUAL TO SECOND DATE" << endl;
            }
            cout << endl;
        }
        if (opt == 4)
        {
            cout << "FIRST DATE" << endl;
            d1.get_date();
            if (d1.valid_date() == false)
            {
                cout << "INVALID" << endl;
                cout << "YOUR DATE IS NOW SET TO 07/01/2000" << endl;
                d1 = date(07, 01, 2000);
                d1.show_date();
                cout << endl << endl;
            }
            cout << "SECOND DATE" << endl;
            d2.get_date();
            if (d2.valid_date() == false)
            {
                cout << "INVALID" << endl;
                cout << "YOUR DATE IS NOW SET TO 07/01/2000" << endl;
                d2 = date(07, 01, 2000);
                d2.show_date();
                cout << endl << endl;
            }
            d2-d1;
        }
        if (opt == 5)
        {
            d1++;
            d1.show_date();
        }
        if (opt == 6)
        {
            --d1;
            d1.show_date();
        }
    } while (opt < 7);

    system("pause");
}
```

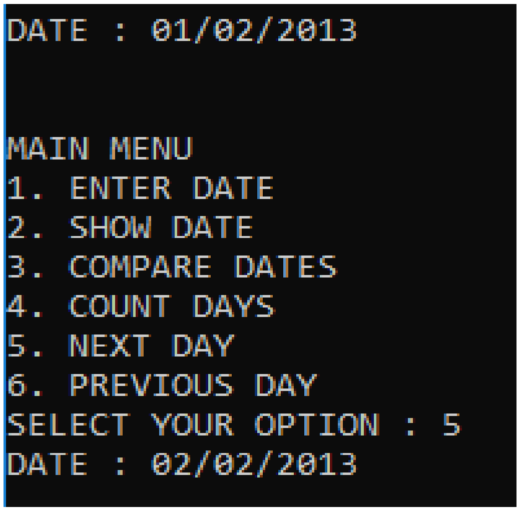
OUTPUT:



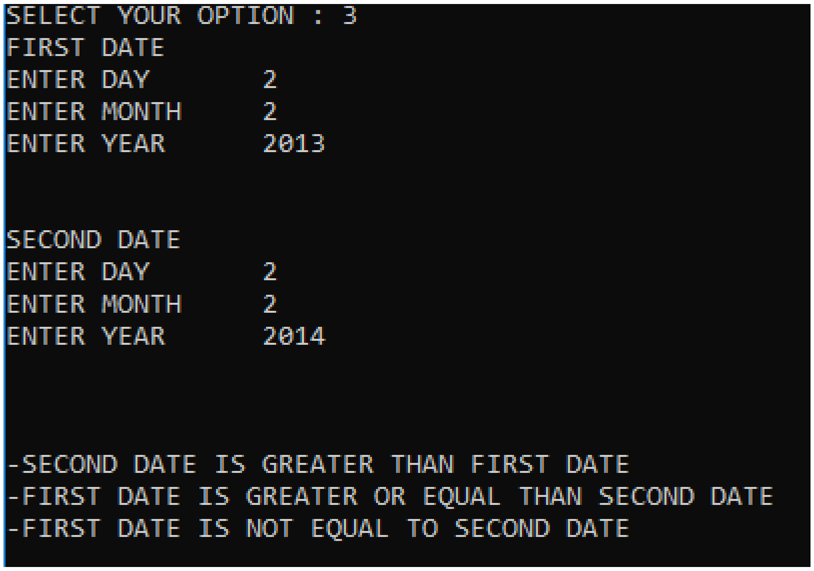
```
SELECT YOUR OPTION : 4
FIRST DATE
ENTER DAY       11
ENTER MONTH     2
ENTER YEAR      2013


SECOND DATE
ENTER DAY       11
ENTER MONTH     2
ENTER YEAR      2014


NO.OF DAYS BETWEEN 2 DATES: 365
```



```
SELECT YOUR OPTION : 1
ENTER DAY          1
ENTER MONTH        2
ENTER YEAR         2013


MAIN MENU
1. ENTER DATE
2. SHOW DATE
3. COMPARE DATES
4. COUNT DAYS
5. NEXT DAY
6. PREVIOUS DAY
SELECT YOUR OPTION : 6
DATE : 31/01/2013
```



```
DATE : 01/02/2013


MAIN MENU
1. ENTER DATE
2. SHOW DATE
3. COMPARE DATES
4. COUNT DAYS
5. NEXT DAY
6. PREVIOUS DAY
SELECT YOUR OPTION : 5
DATE : 02/02/2013
```

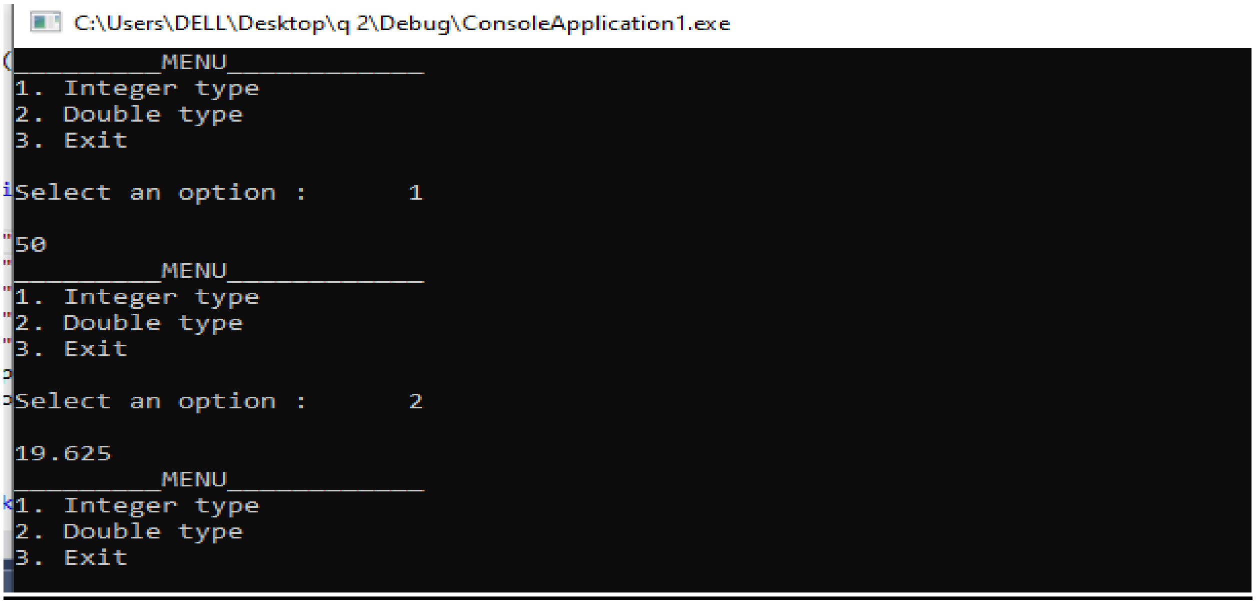


```
SELECT YOUR OPTION : 3
FIRST DATE
ENTER DAY        2
ENTER MONTH      2
ENTER YEAR       2013


SECOND DATE
ENTER DAY        2
ENTER MONTH      2
ENTER YEAR       2014



-SECOND DATE IS GREATER THAN FIRST DATE
-FIRST DATE IS GREATER OR EQUAL THAN SECOND DATE
-FIRST DATE IS NOT EQUAL TO SECOND DATE
```

QUESTION 02:

```
// ConsoleApplication1.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;
template <class t>
t circle_area(t r)
{
        t area;
        area = 3.14*(r*r);
        return area;
}
void main() { double a,b; int op;
do{
                cout << "_________MENU____________\n";
                cout << "1. Integer type\n";
                cout << "2. Double type\n";
                cout << "3. Exit\n";
                cout << " \nSelect an option :\t";
                cin >> op;
                switch (op)
                {
                case 1:a = circle_area(4);
                        cout << endl << a << endl;
                        break;
                case 2:b = circle_area(2.5);
                        cout << endl << b << endl;
                        break;
                case 3:exit(1);
                }
        } while (op < 3);
        system("pause");
}
```

OUTPUT:



C:\Users\DELL\Desktop\q 2\Debug\ConsoleApplication1.exe

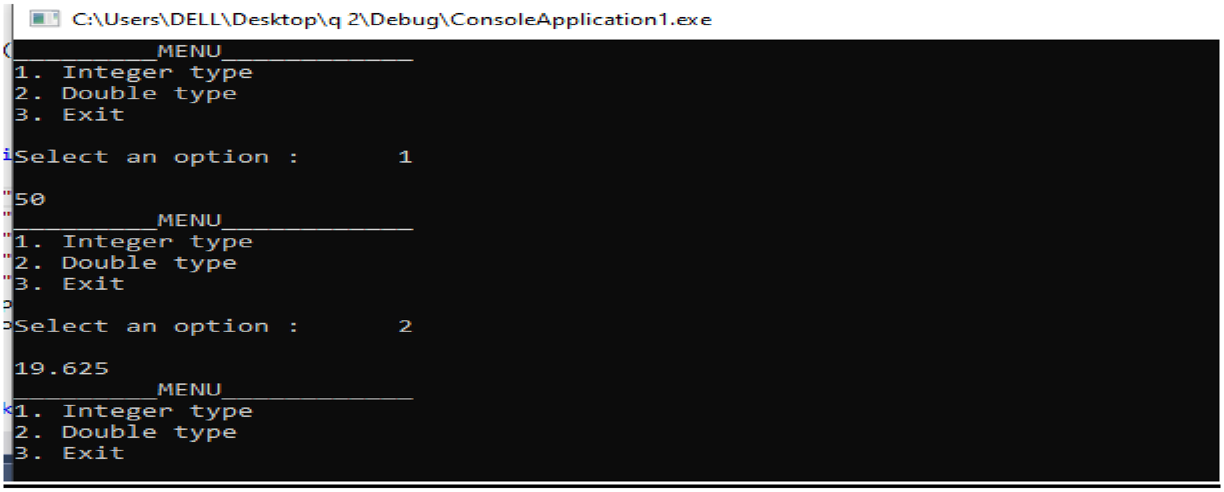
```
__________MENU____________
1. Integer type
2. Double type
3. Exit

Select an option :      1

50
__________MENU____________
1. Integer type
2. Double type
3. Exit

Select an option :      2

19.625
__________MENU____________
1. Integer type
2. Double type
3. Exit
```

QUESTION 03:

```
// ConsoleApplication1.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;
template <class t>
class array
{
private:
t a[10];
public:
       array()
       {
           a[0]=0;
           a[1] = 0;
           a[2] = 0;
       }
       int size;
       void get()
       {
           cout << "enter size : \t";
           cin >> size;
           for (int i = 0; i < size; i++)
           {
               cout << "enter element :\t";
               cin >> a[i];
              }
       }
       void showlist()
       {
              for (int i = 0; i < size; i++)
              {
               cout << a[i] << endl;
              }
       }
       void showfirst()
       {
              cout << a[0] << endl;
       }
       void display(int n)
       {
              cout << a[n] << endl;
       }
       void reverse()
       {
              get();
              int i = size; int j = i - 1;
              i = 0;
              int temp;
              while (i<j)
              {
               temp = a[i];
               a[i] = a[j];
               a[j] = temp;
               i++;
               j--;
              }
              cout << "\nNow the reverse of array is : \n";
               showlist();
       }
       void sum()
       {
              t sum = 0;
              for (int i = 0; i < size; i++)
              {
               sum = sum + a[i];
              }
```

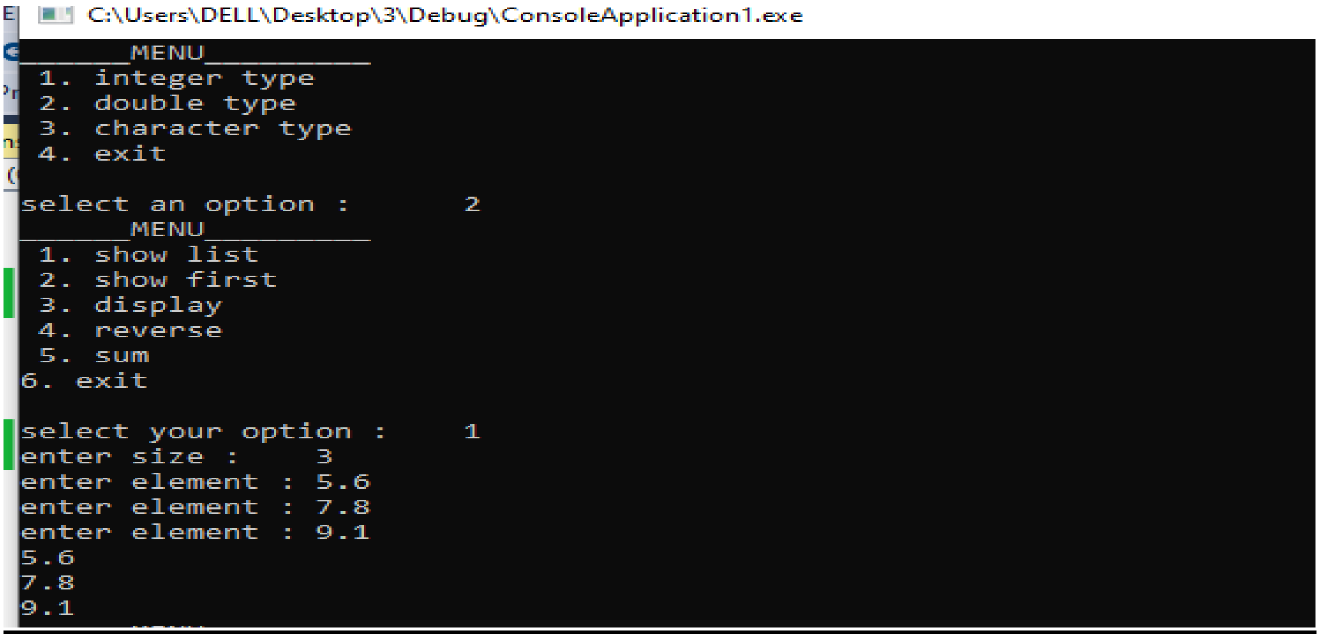
```
              cout << "Sum : \n";
              cout << sum << endl;
       }
};
void main()
{
       array<int>anint;
       array<double>adouble;
       array<char>achar;
       int op; int opt;
       do
       {
              cout << "______MENU_________\n";
              cout << " 1. integer type\n";
              cout << " 2. double type\n";
              cout << " 3. character type\n";
              cout << " 4. exit\n";
              cout << " \nselect an option :\t";
              cin >> op;
              switch (op)
              {
              case 1:do{
                               cout << "______MENU_________\n";
                               cout << " 1. show list\n";
                               cout << " 2. show first\n";
                               cout << " 3. display\n";
                               cout << " 4. reverse\n";
                               cout << " 5. sum\n";
                               cout << "6. exit\n";
                               cout << " \nselect your option :\t";
                               cin >> opt;
                               switch (opt)
                               {
                               case 1:anint.get();
                                      anint.showlist();
                                      break;
                               case 2:anint.showfirst();
                                      break;
                               case 3:anint.display(0);
                                      break;
                               case 4:anint.reverse();
                                      break;
                               case 5:anint.sum();
                                      break;
                               case 6:exit(1);
                               }
              } while (opt < 6);
                     break;
              case 2:do{
                               cout << "______MENU_________\n";
                               cout << " 1. show list\n";
                               cout << " 2. show first\n";
                               cout << " 3. display\n";
                               cout << " 4. reverse\n";
                               cout << " 5. sum\n";
                               cout << "6. exit\n";
                               cout << " \nselect your option :\t";
                               cin >> opt;
                               switch (opt)
                               {
                               case 1:adouble.get();
                                      adouble.showlist();
                                      break;
                               case 2:adouble.showfirst();
                                      break;
                               case 3:adouble.display(1);
                                      break;
                               case 4:adouble.reverse();
                                      break;
                               case 5:adouble.sum();
                                      break;
                               case 6:exit(1);
                               }
              } while (opt < 6);
                               break;
              case 3:do{
                               cout << "______MENU_________\n";
                               cout << " 1. show list\n";
                               cout << " 2. show first\n";
                               cout << " 3. display\n";
                               cout << " 4. reverse\n";
                               cout << " 5. sum\n";
                               cout << "6. exit\n";
                               cout << " \nselect your option :\t";
                               cin >> opt;
                               switch (opt)
                               {
                               case 1:achar.get();
                                      achar.showlist();
                                      break;
                               case 2:achar.showfirst();
                                      break;
                               case 3:achar.display(2);
                                      break;
                               case 4:achar.reverse();
                                      break;
                               case 5:achar.sum();
                                      break;
                               case 6:exit(1);
                               }
              } while (opt < 6);
                     break;
              case 4:exit(1);
              }
       } while (op < 4);
       system("pause");
}
```

OUTPUT:



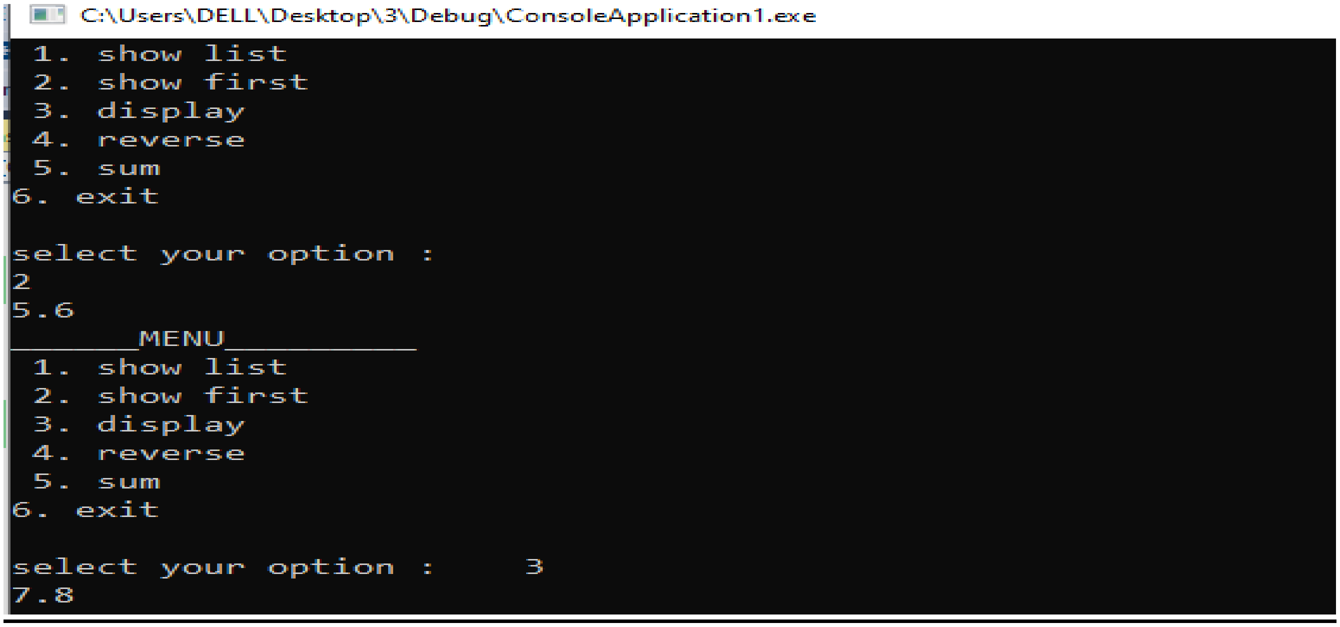
C:\Users\DELL\Desktop\q 2\Debug\ConsoleApplication1.exe

```
_________MENU____________
1. Integer type
2. Double type
3. Exit

Select an option :       1

50
_________MENU____________
1. Integer type
2. Double type
3. Exit

Select an option :       2

19.625
_________MENU____________
1. Integer type
2. Double type
3. Exit
```

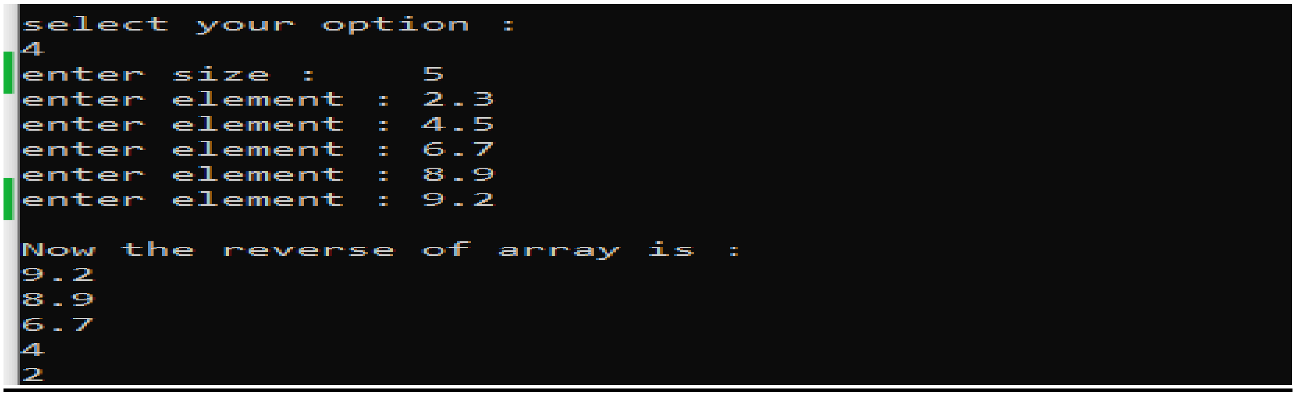


C:\Users\DELL\Desktop\3\Debug\ConsoleApplication1.exe

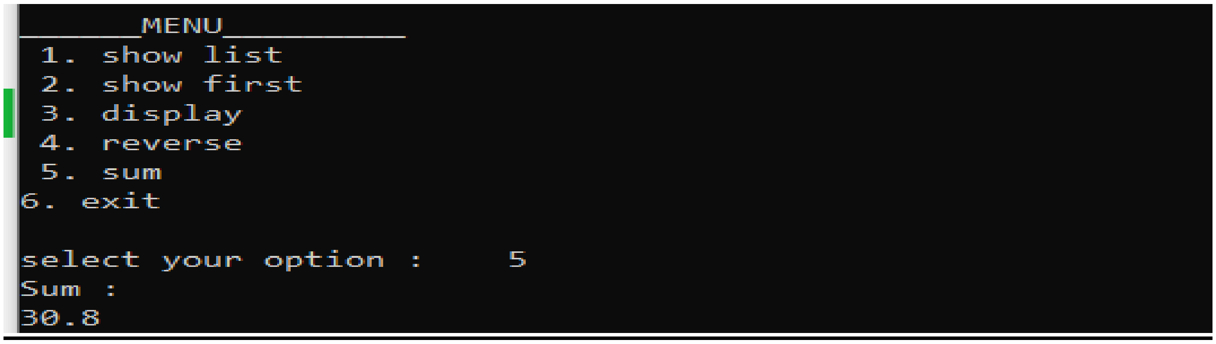
```
______MENU__________
 1. integer type
 2. double type
 3. character type
 4. exit

select an option :       2
______MENU__________
 1. show list
 2. show first
 3. display
 4. reverse
 5. sum
6. exit

select your option :     1
enter size :     3
enter element : 5.6
enter element : 7.8
enter element : 9.1
5.6
7.8
9.1
```



```
C:\Users\DELL\Desktop\3\Debug\ConsoleApplication1.exe
 1. show list
 2. show first
 3. display
 4. reverse
 5. sum
6. exit

select your option :
2
5.6
______MENU__________
 1. show list
 2. show first
 3. display
 4. reverse
 5. sum
6. exit

select your option :     3
7.8
```



```
select your option :
4
enter size :     5
enter element : 2.3
enter element : 4.5
enter element : 6.7
enter element : 8.9
enter element : 9.2

Now the reverse of array is :
9.2
8.9
6.7
4
2
```



```
______MENU__________
 1. show list
 2. show first
 3. display
 4. reverse
 5. sum
6. exit

select your option :     5
Sum :
30.8
```

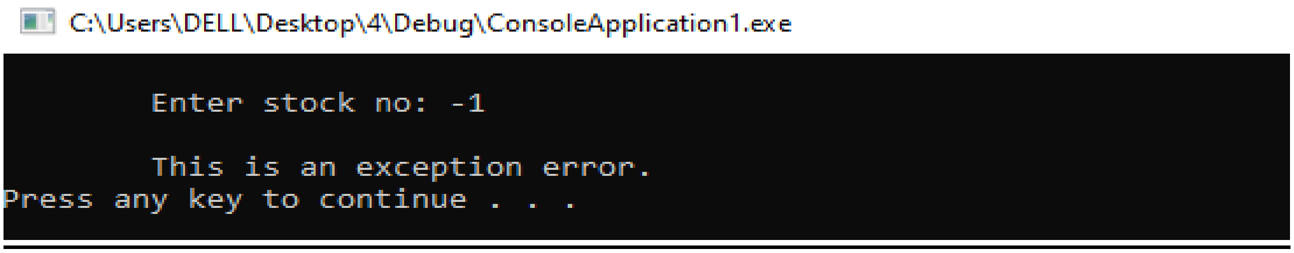
QUESTION 04:

```
// ConsoleApplication1.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
using namespace std;
```

```
class inventory
{
private:
	int stock_no;
	int quantity;
	float price;
public:
	inventory()
	{
		stock_no = 0;
		quantity = 0;
		price = 0;
	}
	class an_excep
	{
	public:
		void error()
		{
			cout << "\n\t\This is an exception error.\n";
		}
	};
	void get()
	{
		cout << "\n\tEnter stock no:\t";
		cin >> stock_no;
		if (stock_no<0 || stock_no>999)
		{
			throw an_excep();
		}
		cout << "\n\tEnter quantity:\t";
		cin >> quantity;
		if (quantity < 0)
		{
			throw an_excep();
		}
		cout << "\n\tEnter price:\t";
		cin >> price;
		if (price>100.00)
		{
			throw an_excep();
		}
	}
	void show()
	{
		cout << "\n\tStock number : " << stock_no << "\n\tQuantity : " << quantity 
<< "\n\tPrice : " << price << endl;
	}
};
void main()
{
	inventory arr[5];
	try{
		for (int i = 0; i < 5; i++)
		{
			arr[i].get();
			cout << endl;
		}
		for (int i = 0; i < 5; i++)
```

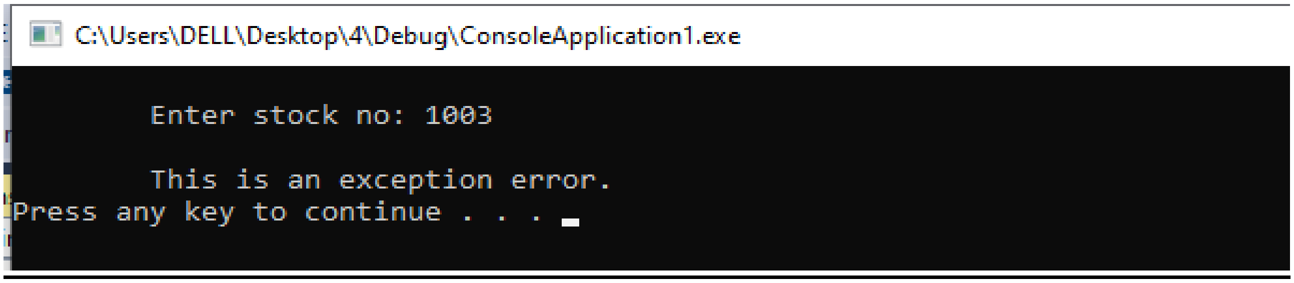
```
        {

                arr[i].show();
                cout << endl;
        }
    }
    catch (inventory::an_excep e)
    {
        e.error();
    }
    system("pause");
}
```

OUTPUT:



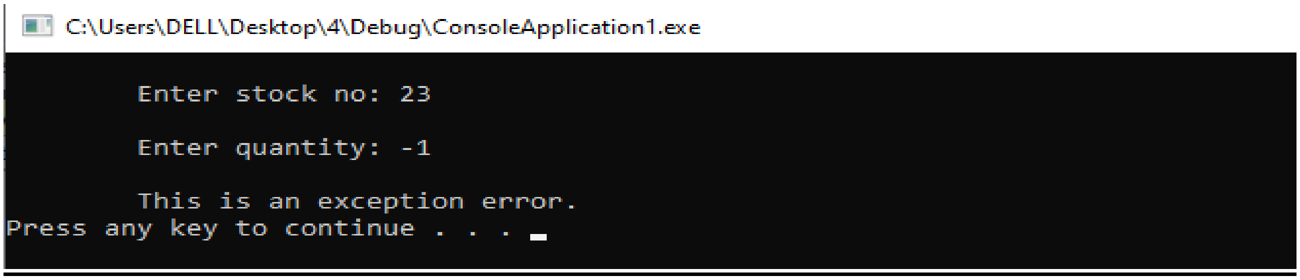
C:\Users\DELL\Desktop\4\Debug\ConsoleApplication1.exe

```
        Enter stock no: -1

        This is an exception error.
Press any key to continue . . .
```



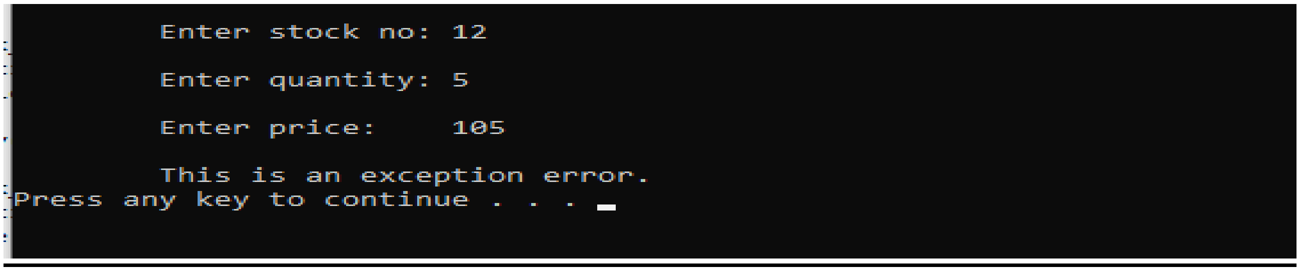
C:\Users\DELL\Desktop\4\Debug\ConsoleApplication1.exe

```
        Enter stock no: 1003

        This is an exception error.
Press any key to continue . . .
```



C:\Users\DELL\Desktop\4\Debug\ConsoleApplication1.exe

```
        Enter stock no: 23

        Enter quantity: -1

        This is an exception error.
Press any key to continue . . .
```



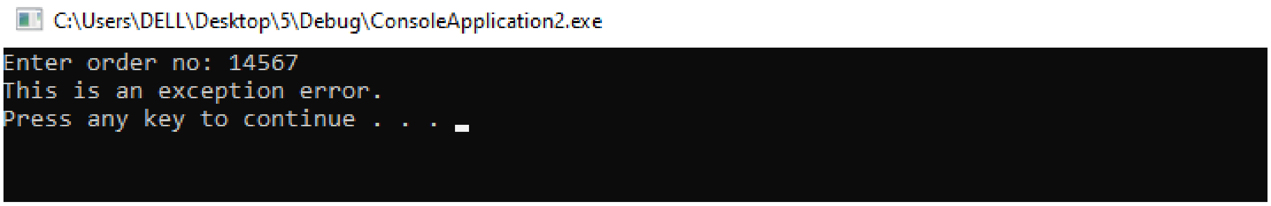
```
        Enter stock no: 12

        Enter quantity: 5

        Enter price:    105

        This is an exception error.
Press any key to continue . . .
```

QUESTION 05:

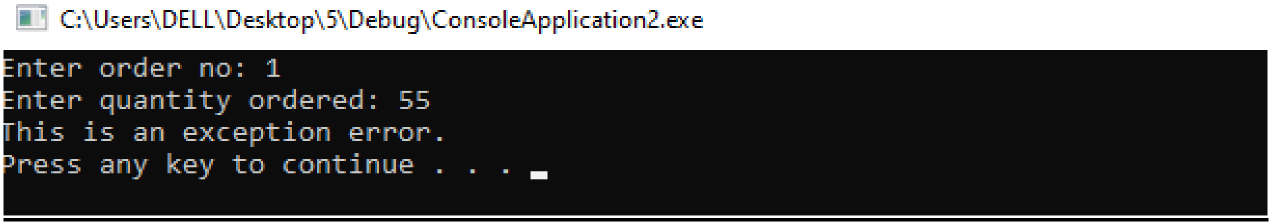
```
// ConsoleApplication2.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include<iostream>
#include<string>
#include<fstream>
using namespace std;
class customer
{
private:
       string order_no;
       int quantity;
       float price;
       float total;
public:
       customer()
       {
              order_no = "0";
              quantity = 0;
              price = 0;
              total = 0;
       }
       customer(string n, int q, float p)
       {
              order_no = n;
              quantity = q;
              price = p;
              total = q*p;
       }
       void get()
       {
              cout << "Enter order no:\t";
              cin >> order_no;
              if (order_no.length() > 4)
              {
                     throw an_excep();
              }
              cout << "Enter quantity ordered:\t";
              cin >> quantity;
              if (quantity > 50)
              {
                     throw an_excep();
              }
              cout << "Enter price:\t";
              cin >> price;
           if (price > 39.95)
              {
                     throw an_excep();
              }
       }
       void show()
       {
              total=quantity*price;
              cout<<"\nTotal : " << total << endl;
```

```
        if (tell_total() > 1000.00)
        {
            throw an_excep();
        }
    }
    class an_excep
    {
    public:
        void error()
        {
            cout << "This is an exception error.\n";
        }
    };
    string tell_order()
    {
        return order_no;
    }
    int tell_quantity()
    {
        return quantity;
    }
    float tell_price()
    {
        return price;
    }
    float tell_total()
    {
        return total;
    }
};
void main()
{
    try{
        customer c[5];
        for (int i = 0; i < 5; i++)
        {
            c[i].get();
            c[i].show();
            cout << endl;
        }

        ofstream ofile("Output.txt");
        for (int i = 0; i < 5; i++)
        {
            ofile << "Order no : " << c[i].tell_order() << endl;
            ofile << "Quantity : " << c[i].tell_quantity() << endl;
            ofile << "Price : " << c[i].tell_price() << endl;
            ofile << "Total : " << c[i].tell_total() << endl;
        }
        cout << "data written on the file\n";
    }
    catch (customer::an_excep e)
    {
        e.error();
    }
    system("pause");

}
```

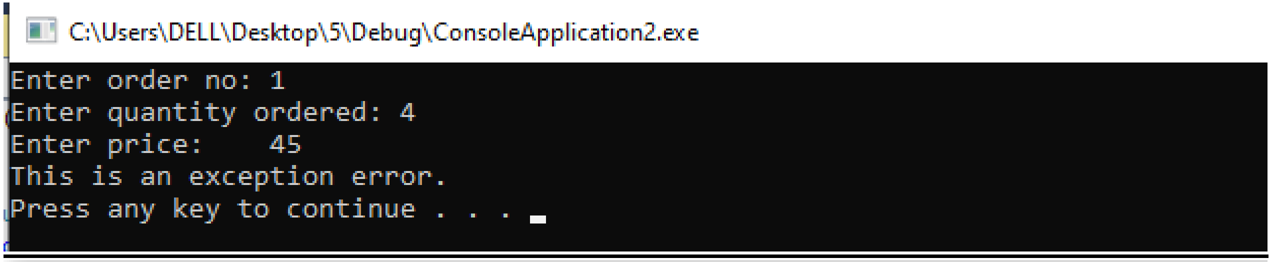
OUTPUT :



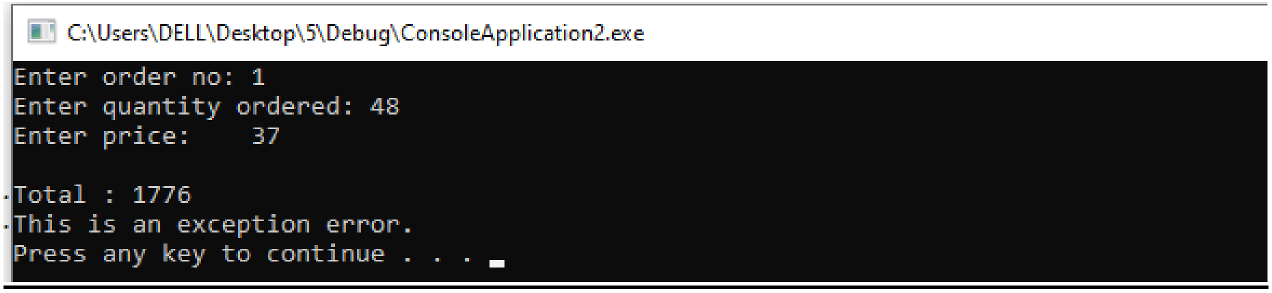
```
Enter order no: 14567
This is an exception error.
Press any key to continue . . .
```



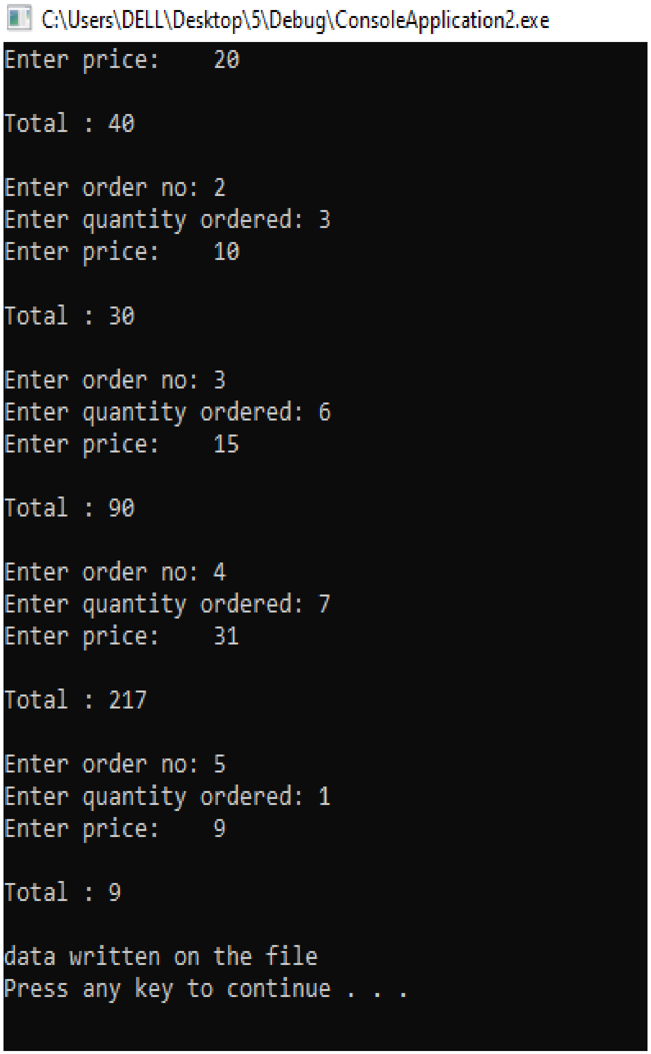
```
Enter order no: 1
Enter quantity ordered: 55
This is an exception error.
Press any key to continue . . .
```



```
Enter order no: 1
Enter quantity ordered: 4
Enter price:    45
This is an exception error.
Press any key to continue . . .
```

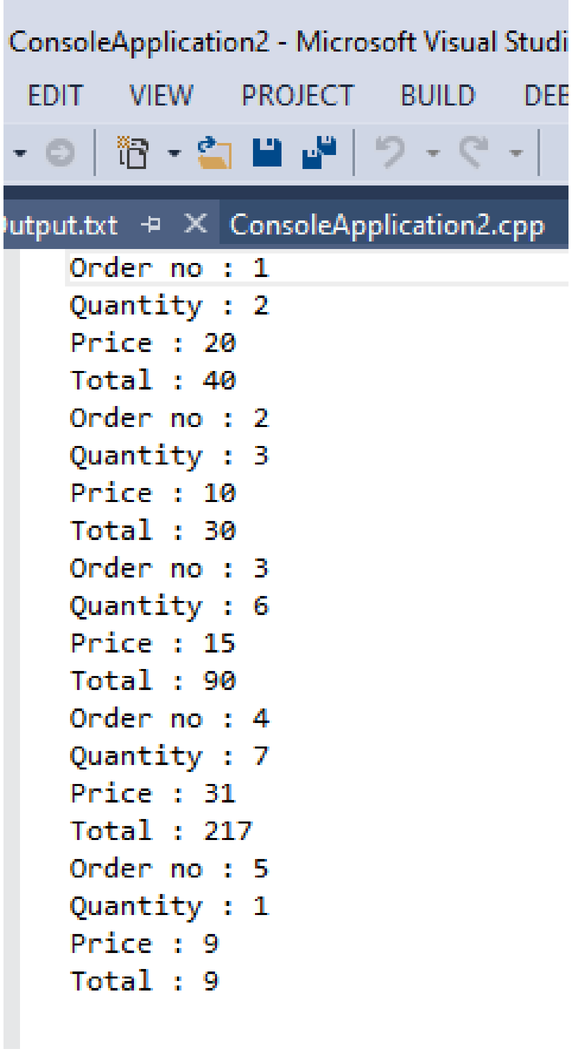


```
Enter order no: 1
Enter quantity ordered: 48
Enter price:    37

Total : 1776
This is an exception error.
Press any key to continue . . .
```



C:\Users\DELL\Desktop\5\Debug\ConsoleApplication2.exe

```
Enter price:    20

Total : 40

Enter order no: 2
Enter quantity ordered: 3
Enter price:    10

Total : 30

Enter order no: 3
Enter quantity ordered: 6
Enter price:    15

Total : 90

Enter order no: 4
Enter quantity ordered: 7
Enter price:    31

Total : 217

Enter order no: 5
Enter quantity ordered: 1
Enter price:    9

Total : 9

data written on the file
Press any key to continue . . .
```



ConsoleApplication2 - Microsoft Visual Studi

```
Order no : 1
Quantity : 2
Price : 20
Total : 40
Order no : 2
Quantity : 3
Price : 10
Total : 30
Order no : 3
Quantity : 6
Price : 15
Total : 90
Order no : 4
Quantity : 7
Price : 31
Total : 217
Order no : 5
Quantity : 1
Price : 9
Total : 9
```