**Experiment No- 7** **Date-**

**Aim – To study Virtual functions and Polymorphism (Runtime Polymorphism)**

## Theory –

**In C++, ios (input-output stream) is a base class for handling console input and output operations. The ios class provides the foundational components for managing input and output streams, offering a wide range of functions and manipulators to customize the formatting and behavior of data streams.**

**Overview of ios in C++**

**The ios class serves as the base class for other stream classes such as istream, ostream, and iostream.**

**It provides various methods to control the formatting of input and output, set error states, and manage stream buffers.**

**Common Member Functions of ios**

**ios::flags**

**. Definition: This function allows getting or setting the format flags for the stream. Format flags determine how input and output operations are performed.**

**ios::setf**

- **Definition**: Sets one or more format flags.

**ios::unsetf**

- **Definition**: Clears the specified format flags.

**ios::precision**

- **Definition**: Sets or retrieves the number of digits displayed for floating-point values.

**ios::width**

- **Definition**: Sets the minimum number of characters to be written or read.

**ios::fill**

- **Definition**: Specifies the character used to fill the width when an output value does not fill the specified field width.
-

## Manipulators in C++

**Manipulators in C++ are special functions that modify the behavior of input and output streams. They are used to format data, control spacing, adjust precision, and manage other**

**stream-related settings. Manipulators provide an easy way to change how data is displayed or read without directly using member functions of the ios class.**

**Commonly Used Manipulators**

1. **endl**
    - **Definition: Inserts a newline character ('\n') into the output stream and flushes the stream. Flushing ensures that all buffered output is written to the console immediately.**

**2 . setw**

- **Definition: Sets the width of the next output field. The output is padded with spaces if the data does not fill the specified width.**

3. **setprecision**

- **Definition: Sets the number of digits to be displayed after the decimal point for floating-point values.**

4. **fixed**

- **Definition: Forces the output of floating-point values to be in fixed-point notation, rather than scientific notation.**

5. **scientific**

- **Definition: Forces the output of floating-point values to be in scientific**

6. **left and right**

- **Definition: Adjust the alignment of output within a field. left aligns the data to the left, while right aligns it to the right (which is the default).**

7. **setfill**

- **Definition: Sets the fill character used when padding output fields that are wider than the data.**

**Manipulators make it easy to control how data is formatted when being output or read. By using manipulators, one can format data to fit specific requirements, such as fixed-width fields, alignment, precision, and display style. This improves code readability and enhances the user experience when interacting with console programs.**

[A] Write a C++ program to print the fol lowing output using ios class member functions(fig-expt4A)

**Program-**

```
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;

class BookTable {
   static const int numRows = 4;
   string names[numRows];
   int codes[numRows];
   float costs[numRows];

public:
   void getData() {
      for (int i = 0; i < 2; i++) {
         cout << "Enter the name of the book: ";
         getline(cin, names[i]);
         cout << "Enter the code: ";
         cin >> codes[i];
         cout << "Enter the cost: ";
         cin >> costs[i];
         cin.ignore();
      }

      for (int i = 2; i < numRows; i++) {
         names[i] = "";
         codes[i] = 0;
         costs[i] = 0.0;
      }
   }

   void displayTable() {
      cout << "-----------------------------------------\n";
```

OUTPUT –

```
Enter the name of the book: Turbo C++
Enter the code: 1001
Enter the cost: 250.95
Enter the name of the book: C Primer
Enter the code: 905
Enter the cost: 95.70
-----------------------------------------
| NAME              | CODE    |     COST |
-----------------------------------------
| Turbo C++         | 1001    |   250.95 |
| C Primer          | 905     |    95.70 |
| .....             | .....   |     0.00 |
| .....             | .....   |     0.00 |
-----------------------------------------
```

```
    cout << "| " << left << setw(18) << "NAME" << "| " << setw(8) << "CODE" << "| " << right << setw(8) << "COST" << " |" << endl;
    cout << "-----------------------------------------\n";

    for (int i = 0; i < numRows; i++) {
        cout << "| " << left << setw(18) << (names[i].empty() ? "....." : names[i])
             << "| " << setw(8) << (codes[i] == 0 ? "....." : to_string(codes[i]))
             << "| " << right << fixed << setw(8) << setprecision(2) << (costs[i] == 0.0 ? 0.0 : costs[i]) << " |" << endl;
    }

    cout << "-----------------------------------------\n";
  }
};

int main() {
  BookTable table;
  table.getData();
  table.displayTable();

  return 0;
}
```

## [C] Write a C++ program to format the following o/p using manipulators(fig-expt4C)

**Program –**

```
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;

class BookTable {
   static const int numRows = 4;
   string names[numRows];
   int codes[numRows];
   float costs[numRows];

public:
   void getData() {
      for (int i = 0; i < 2; i++) {
         cout << "Enter the name of the book: ";
         getline(cin, names[i]);
         cout << "Enter the code: ";
         cin >> codes[i];
         cout << "Enter the cost: ";
         cin >> costs[i];
         cin.ignore();
      }

      for (int i = 2; i < numRows; i++) {
         names[i] = "";
         codes[i] = 0;
         costs[i] = 0.0;
      }
   }

   void displayTable() {
      cout << "-----------------------------------------------
--\n";
      cout << "| " << left << setw(20) << "NAME"
           << "| " << setw(10) << "CODE"
           << "| " << right << fixed << setw(10) <<
setprecision(2) << "COST" << " |\n";
      cout << "-----------------------------------------------
--\n";

      for (int i = 0; i < numRows; i++) {
            cout << "| " << left << setw(20) <<
(names[i].empty() ? "....." : names[i])
            << "| " << setw(10) << (codes[i] == 0 ?
"....." : to_string(codes[i]))
                  << "| " << right << setw(10) <<
setprecision(2) << (costs[i] == 0.0 ? 0.00f :
costs[i]) << " |\n";
```

**Output –**

```
Enter the name of the book: Turbo C++
Enter the code: 1001
Enter the cost: 250.95
Enter the name of the book: C Primer
Enter the code: 905
Enter the cost: 95.70
-----------------------------------------------
| NAME                | CODE       |       COST |
-----------------------------------------------
| Turbo C++           | 1001       |     250.95 |
| C Primer            | 905        |      95.70 |
| .....               | .....      |       0.00 |
| .....               | .....      |       0.00 |
-----------------------------------------------
```

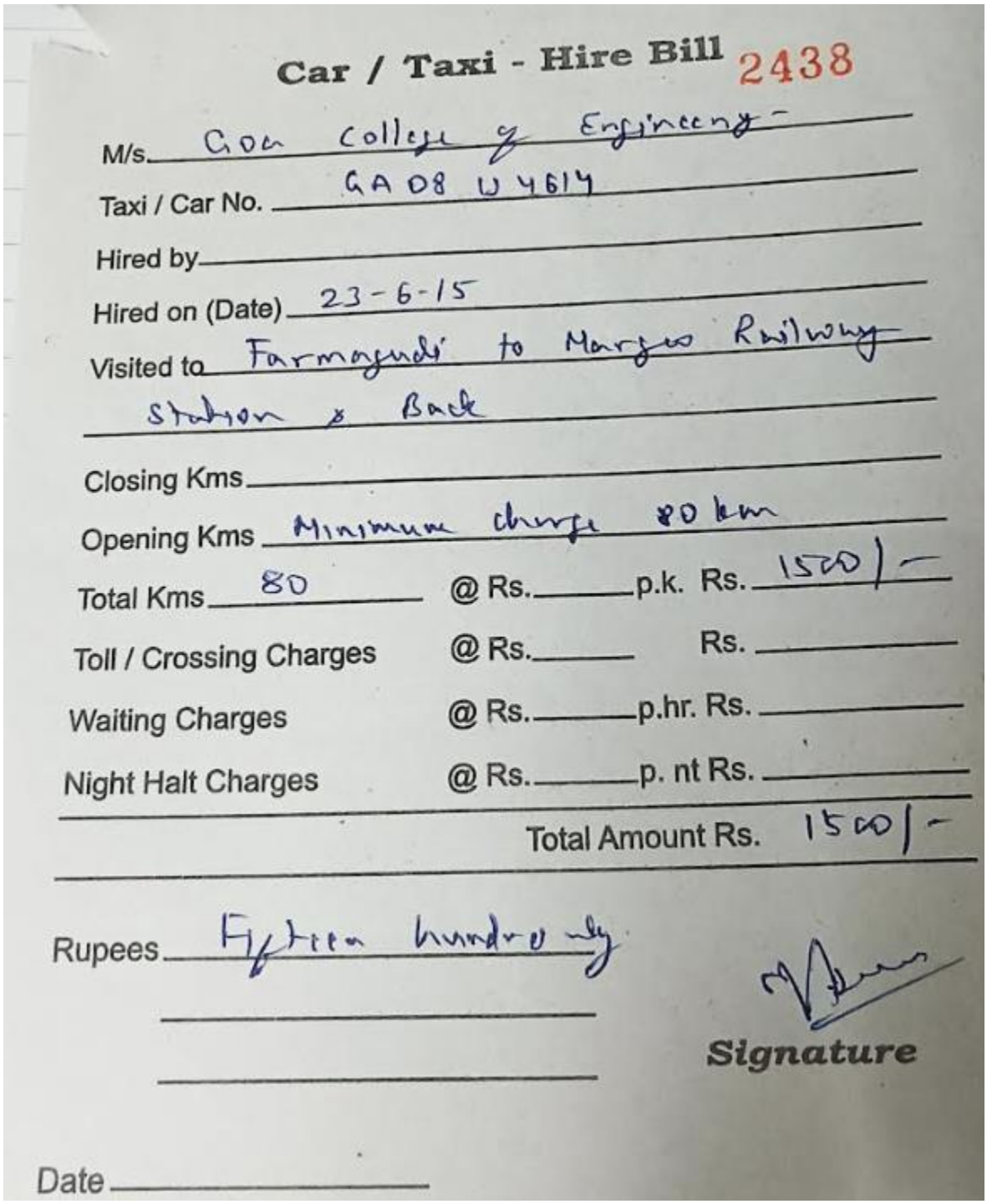
```
    }

    cout << "-----------------------------------------------
--\n";
  }
};

int main() {
  BookTable table;
  table.getData();
  table.displayTable();

  return 0;
}
```

**[D]Write a C++ code to replicate the table given below using the concepts of Manipulators**



Car / Taxi - Hire Bill 2438

M/s. Goa College of Engineering

Taxi / Car No. GA 08 U 4614

Hired by

Hired on (Date) 23-6-15

Visited to Farmagudi to Margao Railway Station & Back

Closing Kms

Opening Kms Minimum charge 80 km

Total Kms 80 @ Rs. p.k. Rs. 1500/-

Toll / Crossing Charges @ Rs. Rs.

Waiting Charges @ Rs. p.hr. Rs.

Night Halt Charges @ Rs. p. nt Rs.

Total Amount Rs. 1500/-

Rupees Fifteen hundred only

Signature

Date

## Program –

```
#include <iostream>
#include <iomanip>
#include <string>
#include <vector>
#include <ctime>
#include <cstdlib>
#include <sstream>
#define underline "\033[4m"
using namespace std;

string singleDigit[] = {"", "one", "two", "three", "four",
"five", "six", "seven", "eight", "nine"};
string twoDigits[] = {"", "ten", "eleven",
 "twelve", "thirteen", "fourteen",
 "fifteen", "sixteen", "seventeen", "eighteen", "nineteen"};
string tensMultiple[] = {"", "", "twenty", "thirty", "forty",
"fifty", "sixty", "seventy", "eighty", "ninety"};
string powerOfTen[] = {"", "thousand", "million", "billion"};

// Function to convert numbers less than 1000 to words
string convertToWordsBelowThousand(int n) {
   string result = "";

   if (n >= 100) {
      result += singleDigit[n / 100] + " hundred";
      n %= 100;
      if (n) result += " and ";
   }

   if (n >= 20) {
      result += tensMultiple[n / 10];
      n %= 10;
      if (n) result += " " + singleDigit[n];
   } else if (n >= 10) {
      result += twoDigits[n % 10];
   } else if (n > 0) {
      result += singleDigit[n];
   }

   return result;
}

// Function to convert a number to words
string convertToWords(int n) {
   if (n == 0) return "zero";

   string result = "";
   int count = 0;

   while (n > 0) {
      int part = n % 1000;
      if (part > 0) {
                                    string partWord =
convertToWordsBelowThousand(part);
```

## Output –

```
Enter the number of people you want to make the bill for

1
====================Constumer 1=======================


===== Car/Taxi Hire Bill Menu =====
1. Input Bill Details
2. Display Bill Details
3. Exit
Enter your choice: 1
Enter customer name: Ramesh Naik
Enter car number: GA O7 T 7876
Enter hiring person name: RAHUL
Enter hiring date: 10/10/24
Enter visiting route: Panaji To Madgao
Enter closing kilometer : 100 per km
Enter opening kilometer: 80 for first 25
Enter closing kilometer and charge per kilometer
10
100
Enter opening kilometer and charge per kilometer
25
80
Enter toll charge: 78
Enter waiting charge per hour: 90
Enter total waiting time (in hours): 2
Enter night charge per hour: 899
Enter total nights: 2


===== Car/Taxi Hire Bill Menu =====
1. Input Bill Details
2. Display Bill Details
3. Exit
Enter your choice: 2
```

```
                                NARAYAN TRANSPORTS
                                Car / Taxi - Hire Bill
                                                        7648

 M/s __________Ramesh Naik_____________________________________________
Taxi/Car No._____________GA O7 T 7876_____________________________________
Hired by._____RAHUL_____________________________________________________
Hired on (Date)-_________10/10/24_____________________________________
Visited to____Panaji To Madgao_____________________________________

Closing Kms.__100 per km_____________________________________________
Opening Kms_________80 for first 25_____________________________________
Total kms_________35_______ @ Rs._____________85.71 p.k.       Rs._____3000.00
Toll/Crossing Charges        @ Rs._____________78.00      Rs._____________78.00
Waiting Charges              @ Rs._____________90.00 p.h.r    Rs._____________180.00
Night Halt Charges           @ Rs._____________899.00 p.nt    Rs._____________1798.00

                                                        Total -> 5056

Rupees - five thousand fifty six                                        (T.B)
         Time: 15-10-2024 07:37:49                                   SIGNATURE
```

```
===== Car/Taxi Hire Bill Menu =====
1. Input Bill Details
2. Display Bill Details
3. Exit
Enter your choice: 3
Exiting the program.
```

```
            if (count > 0 && partWord != "") partWord += " " +
powerOfTen[count];
            result = partWord + (result.empty() ? "" : " ") + result;
        }
        n /= 1000;
        count++;
    }

    return result;
}

int generateRandomNumber() {
    srand(time(0)); // Seed the random number generator
with the current time
    return 1000 + rand() % 9000; // Ensures the number is
between 1000 and 9999
}

string getCurrentDateTime() {
    time_t now = time(0);
    tm *ltm = localtime(&now);

    ostringstream oss;
    oss << setw(2) << setfill('0') << ltm->tm_mday << "-"
        << setw(2) << setfill('0') << 1 + ltm->tm_mon << "-"
        << 1900 + ltm->tm_year << " "
        << setw(2) << setfill('0') << ltm->tm_hour << ":"
        << setw(2) << setfill('0') << ltm->tm_min << ":"
        << setw(2) << setfill('0') << ltm->tm_sec;

    return oss.str();
}

class Bill {
public:
    long int bill_num = 2437;
    string customer;
    int BILL_N ;
    string carnumber;
    string hiringperson;
    string hiring_date;
    string visiting_route;
    string closing_kilometer;
    string opening_kilometer;
    int close_kilo ;
    float close_charge ;
    int open_kilo ;
    float open_charge ;
    int total_kilometer;
    float charge_perkm;
    float total_charge;
    float toll_charge;
    float waiting_chargeperhr;
    int total_tym;
    float Total_wait_charge;
```

```
      cout << "Enter waiting charge per hour: ";
      cin >> waiting_chargeperhr;

      cout << "Enter total waiting time (in hours): ";
      cin >> total_tym;

      Total_wait_charge = (total_tym * waiting_chargeperhr);

      cout << "Enter night charge per hour: ";
      cin >> night_chargeperhr;

      cout << "Enter total nights: ";
      cin >> total_nights;

     Total_night_charge = (night_chargeperhr * total_nights);

      total = total_charge + toll_charge + Total_wait_charge +
Total_night_charge;


      charge_in_words = convertToWords(total);

     time = getCurrentDateTime();
   }

   void displayDetails() {
   cout << setw(104) << setfill('_') << "" << endl;
   cout << "|" << setw(102) << setfill(' ') << "" << "|" << endl;

    cout << "|" << setw(38) << setfill(' ') << "" << "NARAYAN
TRANSPORTS" << setw(46) << setfill(' ') << "" << "|" << endl;

   cout << "|" << setw(38) << setfill(' ') << "" << "Car / Taxi -
Hire Bill" << setw(42) << setfill(' ') << "" << "|" << endl;

    cout << "|" << setw(64) << setfill(' ') << "" << right <<
bill_num << setw(34) << setfill(' ') << "" << "|" << endl;

   cout << "|" << setw(102) << setfill(' ') << "" << "|" << endl;

    cout << "|" << setw(15) << setfill(' ') << "" << left <<
setw(15) << setfill('_') << "M/s "
       << underline << setw(60) << setfill('_') << customer <<
"\033[0m" << setw(12) << setfill(' ') << "" << "|" << endl;

    cout << "|" << setw(14) << setfill(' ') << "" << left <<
setw(25) << setfill('_') << "Taxi/Car No."
       << underline << setw(50) << setfill('_') << carnumber <<
"\033[0m" << setw(13) << setfill(' ') << "" << "|" << endl;

    cout << "|" << setw(14) << setfill(' ') << "" << left <<
setw(15) << setfill('_') << "Hired by."
       << underline << setw(60) << setfill('_') << hiringperson
<< "\033[0m" << setw(13) << setfill(' ') << "" << "|" << endl;
```

```
    cout << "|" << setw(14) << setfill(' ') << "" << left <<
setw(25) << setfill('_') << "Hired on (Date)-"
      << underline << setw(50) << setfill('_') << hiring_date <<
"\033[0m" << setw(13) << setfill(' ') << "" << "|" << endl;

    cout << "|" << setw(14) << setfill(' ') << "" << left <<
setw(15) << setfill('_') << "Visited to"
      << underline << setw(60) << setfill('_') << visiting_route
<< "\033[0m" << setw(13) << setfill(' ') << "" << "|" << endl;

   cout << "|" << setw(102) << setfill(' ') << "" << "|" << endl;

    cout << "|" << setw(14) << setfill(' ') << "" << left <<
setw(15) << setfill('_') << "Closing Kms."
           << underline << setw(60) << setfill('_') <<
closing_kilometer << "\033[0m" << setw(13) << setfill(' ') <<
"" << "|" << endl;

    cout << "|" << setw(14) << setfill(' ') << "" << left <<
setw(20) << setfill('_') << "Opening Kms"
           << underline << setw(55) << setfill('_') <<
opening_kilometer << "\033[0m" << setw(13) << setfill(' ') <<
"" << "|" << endl;

    cout << "|" << setw(14) << setfill(' ') << "" << left <<
setw(20) << setfill('_') << "Total kms"
           << underline << setw(10) << setfill('_') <<
total_kilometer << "\033[0m" << setw(8) << setfill('_')
      << " @ Rs." << setw(15) << setfill('_') <<setprecision(2)
<< fixed
       << underline << charge_perkm << "\033[0m"
      << setw(11)<< setfill(' ') << left << " p.k." << setw(2) <<
setfill(' ') << left << "  Rs."
         << setw(9) << setfill('_') << underline << fixed
<<setprecision(2) <<total_charge << "\033[0m" << setw(5)
<< setfill(' ') << "" << "|" << endl;

    cout << "|" << setw(14) << setfill(' ') << "" << left <<
setw(30) << "Toll/Crossing Charges"
      << setw(8) << setfill('_') << " @ Rs." << setw(15)<< fixed
<<setprecision(2) << setfill('_') << underline << toll_charge <<
"\033[0m"
     << setw(4) <<setfill(' ') << left << " " << setw(5) << setfill('
') << left << "  Rs."
        << setw(15)<< fixed <<setprecision(2) << setfill('_')
<<underline<< toll_charge<<"\033[0m" << setw(9) << setfill('
') << "" << "|" << endl;

    cout << "|" << setw(14) << setfill(' ') << "" << left <<
setw(30) << "Waiting Charges"
      << setw(7) << setfill('_') << " @ Rs." << setw(13) << fixed
<<setprecision(2)<< setfill('_') << underline <<
waiting_chargeperhr << "\033[0m"
      << setw(10) <<setfill(' ')<< left << " p.h.r " << setw(3) <<
setfill('_') << left << "  Rs."
```

```
        << setw(15) << fixed <<setprecision(2) << setfill('_')
<<underline<< Total_wait_charge <<"\033[0m" << setw(5)
<< setfill(' ') << "" << "|" << endl;

    cout << "|" << setw(14) << setfill(' ') << "" << left <<
setw(30) << "Night Halt Charges"
       << setw(7) << setfill('_') << " @ Rs." << setw(15)<< fixed
<<setprecision(2) << setfill('_') << underline <<
night_chargeperhr << "\033[0m"
       << setw(5) << left << " p.nt " << setw(5) << setfill('_') <<
left << "  Rs."
         << setw(15)<< fixed <<setprecision(2) << setfill('_')
<<underline<< Total_night_charge <<"\033[0m"<< setw(5)
<< setfill(' ') << "" << "|" << endl;

   cout << "|" << setw(102) << setfill(' ') << "" << "|" << endl;

   cout << "|" << setw(14) << setfill(' ') << "" << setw(75) <<
setfill('_') << ""
       << setw(13) << setfill(' ') << "" << "|" << endl;

   cout << "|" << setw(14) << setfill(' ') << "" << setw(50) <<
setfill(' ')
       << "" << "Total -> " << setw(19) << total << setw(10) <<
setfill(' ') << "" << "|" << endl;

   cout << "|" << setw(14) << setfill(' ') << "" << setw(75) <<
setfill('_') << ""
       << setw(13) << setfill(' ') << "" << "|" << endl;

   cout << "|" << setw(102) << setfill(' ') << "" << "|" << endl;

   cout << "|" << setw(14) << setfill(' ') << "" << left << setw(6)
       << "Rupees - " << underline << setw(59) << setfill(' ') <<
left << charge_in_words << "\033[0m" << setw(12)
       << setfill(' ') << "" << setw(6) << setfill(' ') << " (T.B) " <<
setw(1) << setfill(' ') << "" << "|" << endl;

    cout << "|" << setw(24) << setfill(' ') << "" << "Time: " <<
underline << time << "\033[0m" << setw(15)
       << setfill(' ') << "" << setw(37) <<right<< setfill(' ') << "
SIGNATURE  " << setw(1) << setfill(' ') << "" << "|" << endl;

   cout << "|" << setw(102) << setfill(' ') << "" << "|" << endl;

   cout << "|" << setw(102) << setfill('_') << "" << "|" << endl;
}

};

int main() {

   // #ifndef ATHARV_CUSTOM
   // freopen("in.txt","r",stdin);
   // #endif
```

```
   Bill bill;
   char choice;
   int n ;

   cout<<"Enter the number of people you want to make the
bill for\n"<<endl ;
   cin>>n ;
for (int i = 0; i < n; i++) {
   Bill bill;
    cout  <<"====================="<<"Constumer  "<<i+1
<<"========================";
   cout <<endl ;
   do {
      cout << "\n\n===== Car/Taxi Hire Bill Menu =====\n";
      cout << "1. Input Bill Details\n";
      cout << "2. Display Bill Details\n";
      cout << "3. Exit\n";
      cout << "Enter your choice: ";
      cin >> choice;
      cin.ignore();

      switch (choice) {
         case '1':
            bill.inputDetails();
            break;
         case '2':
            bill.displayDetails();
            break;
         case '3':
            cout << "Exiting the program. \n \n \n\n";
            break;
         default:
            cout << "Invalid choice. Please try again.\n";
            break;
      }
   } while (choice != '3');
}


   return 0;
}
```

**Conclusion – All the codes were successfully executed using the concepts of console I/O**

**[B] Write a C++ program which reads a text from keyboard and display the following**

**information on screen in 3 column format**

**1. Number of lines**

**2. Number of words**

**3. Number of characters**

**Strings should be left justified and numbers to be right justified. Use suitable field width**

**Program –**

```
#include <iostream>
#include <iomanip>
#include <string>
#include <sstream>

using namespace std;

class TextAnalyzer {
private:
   int lineCount;
   int wordCount;
   int charCount;

public:
        TextAnalyzer() : lineCount(0), wordCount(0), charCount(0) {}

   void analyzeText() {
      string inputText;
        cout << "Enter text (type 'END' on a new line to finish):" << endl;

      while (true) {
         getline(cin, inputText);
         if (inputText == "END") {
            break;
         }

         lineCount++;
         charCount += inputText.length();

         stringstream ss(inputText);
         string word;
         while (ss >> word) {
            wordCount++;
         }
      }
   }

   void displayResults() const {
      cout << "\n" << setfill('-') << setw(38) << "-" << setfill(' ') << endl;
      cout << "|"<< left << setw(25) << "Number of lines"
```

**OUTPUT –**

```
Enter text (type 'END' on a new line to finish):
Hello
Konichiwa
Ni Hao
Hola
END

--------------------------------------
|Number of lines                   4 |
--------------------------------------
|Number of words                   5 |
--------------------------------------
|Number of characters             25 |
|-------------------------------------
```

```
        << right << setw(10) << lineCount << " |" << endl;

      cout << setfill('-') << setw(38) << "-" << setfill(' ') << endl;
      cout <<"|"<< left << setw(25) << "Number of words"
         << right << setw(10) << wordCount << " |" << endl;

      cout << setfill('-') << setw(38) << "-" << setfill(' ') << endl;
         cout <<"|"<< left << setw(25) << "Number of characters"
         << right << setw(10) << charCount << " |" << endl;
      cout <<"|"<< setfill('-') << setw(60) << "-" << setfill(' ') << endl;
   }
};

int main() {
   TextAnalyzer analyzer;
   analyzer.analyzeText();
   analyzer.displayResults();
   return 0;
}
```