

Programming Laboratory-I

Assignment No-6

(Stream classes and File handling)

PRN : 2020BTECS00005

NAME: JADHAV SANKET JADHAV.

1. Write a program to read a list containing item name, item code, and cost interactively and produce a three column output as shown below.

NAME	CODE	COST
Turbo C++	1001	250.95
C Primer	905	95.70

Note that the name and code are left justified and the cost is right justified with a precision of two digits. Trailing zeros are shown.

CODE:

A screenshot of a Visual Studio Code editor window. The title bar shows '1.cpp - Lab - Visual Studio Code'. The editor contains a C++ program. The code defines a class 'item' with attributes 'name' (char array), 'code' (int), and 'cost' (float). It has a 'get_data' method to read input and a 'display' method to output the data in a formatted three-column table. The 'display' method uses 'cout' with various formatting flags like 'precision(2)', 'fixed', 'floatfield', 'showpoint', 'left', 'adjustfield', 'setw(40)', 'right', and 'setw(15)' to achieve the desired output format. The code is as follows:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  class item
5  {
6      char name[40];
7      int code;
8      float cost;
9  public:
10     void get_data(char *n,int c,float co)
11     {
12         strcpy(name,n);
13         code=c;
14         cost=co;
15     }
16     void display();
17 };
18
19
20 void item:: display()
21 {
22     cout.precision(2);
23     cout.setf(ios::fixed,ios::floatfield);
24     cout.setf(ios::showpoint);
25     cout.setf(ios::left,ios::adjustfield);
26     cout<<setw(40)<<name<<code;
27     cout.setf(ios::right,ios::adjustfield);
28     cout<<setw(15)<<cost<<endl;
29 }
```

```
int main()
{int i;
    item a[5];
    a[0].get_data("Tarbo C++",1001,250.95);
    a[1].get_data("C primer",905,95.7);
    a[2].get_data("algorithm",1111,120.5);
    a[3].get_data("principle of electronics",2220,150.85);
    a[4].get_data("solution of balagurusamy",6666,145.00);
    cout<<setw(10)<<"name"<<setw(34)<<"code"<<setw(15)<<"cost"<<endl;
    for(int i=0;i<60;i++)
        cout<<"-";
    cout<<endl;
    for(i=0;i<5;i++)
        a[i].display();
    return 0;
}
```

OUTPUT:

name	code	cost
<hr/>		
Tarbo C++	1001	250.95
C Primer	905	95.70
Algorithm	1111	120.50
Principle of electronics	2220	150.85
Solution of balaguruswamy	6666	145.00

2. Modify the above program to fill the unused spaces with hyphens.

CODE:

```

2.cpp 1 x
assignment_6 > 2.cpp > display()
1  #include<iostream>
2  #include<iomanip>
3  #include<string>
4  using namespace std;
5  class item
6  {
7      char name[40];
8      int code;
9      float cost;
10     public:
11     void get_data(char *n,int c,float co)
12     {
13         strcpy(name,n);
14         code=c;
15         cost=co;
16     }
17     void display();
18 };
19 void item:: display()
20 {
21     cout.precision(2);
22     cout.fill('-');

```

```

2.cpp 1 x
assignment_6 > 2.cpp > display()
26     cout.setf(ios::showpoint);
27     cout.setf(ios::left,ios::adjustfield);
28     cout<<setw(40)<<name<<code;
29     cout.setf(ios::right,ios::adjustfield);
30     cout<<setw(15)<<cost<<endl;
31 }
32
33 int main()
34 {
35     item a[5];
36     a[0].get_data("Tarbo C++",1001,250.95);
37     a[1].get_data("C primer",905,95.7);
38     a[2].get_data("algorithm",1111,120.5);
39     a[3].get_data("principle of electronics",2220,150.85);
40     a[4].get_data("solution of balagurusamy",6666,145.00);
41     cout<<setw(10)<<"name"<<setw(34)<<"code"<<setw(15)<<"cost"<<endl;
42     for(int i=0;i<60;i++)
43         cout<<"-";
44     cout<<endl;
45     int i;
46     for(i=0;i<5;i++)
47         a[i].display();
48     return 0;
49 }

```

OUTPUT:

name	code	cost
------	------	------

Tarbo C++	1001	250.95
C Primer	905	95.70
algorithm	1111	120.50
Principle of electronics	2220	150.85
Solution of balaguruswamy	6666	145.00

3. Write a program which reads a text from the keyboard and displays the following information on the screen in two columns:
- (a) Number of lines
 - (b) Number of words
 - (c) Number of characters

Strings should be left-justified and numbers should be right-justified in a suitable field width.

CODE:

3.cpp 2 x

Assignment_6 > 3.cpp > main()

```
1  #include<iostream>
2  #include<iomanip>
3  #include<string>
4  #include<bits/stdc++.h>
5  using namespace std;
6  int main()
7  {
8      char line[1000];
9      char ch;
10     int c;
11     int word,lines,chr;
12     word=0;
13     lines=0;
14     chr=0;
15     int end=0;
16     cout<<" Enter text : \n";
17     while(end==0)
18     {
19         c=0;
20         while((ch=getchar())!='\n')
21             line1=ch;
22         line1='\0';
23         if(line[0]!='\0')
24             break;
25         else
```

3.cpp 2 x

Assignment_6 > 3.cpp > main()

```
24         break;
25         else
26         {
27             word++;
28             for(int i=0;line[i]!='\0';i++)
29                 if(line[i]==' ' || line[i]=='\t' || line[i]=='\n')
30                     word++;
31             }
32             lines++;
33             chr+=strlen(line);
34         }
35
36         cout.setf(ios::left,ios::adjustfield);
37         cout<<setw(25)<<"Number of lines"<<setw(25)
38         <<"Number of words "<<"Number of characters "<<endl;
39         cout.setf(ios::right,ios::adjustfield);
40         cout<<setw(10)<<lines<<setw(24)<<word<<setw(25)<<chr<<endl<<endl;
41         return 0;
42     }
```

OUTPUT:

santo reads a poem.

He is good person human.

He respects his teachers.

He feels shy when I admire him.

I like his morality.

No. of lines	No.of words	No.of characters
--------------	-------------	------------------

5	25	128
---	----	-----

4. Write a program that reads a text file and creates another file that is identical except that every sequence of consecutive blank spaces is replaced by a single space.

CODE:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main()
5  {
6      char ch;
7      int count = 0;
8      ifstream in_stream;
9      ofstream out_stream;
10
11     in_stream.open("A.txt");
12     out_stream.open("B.txt");
13     while (!in_stream.eof())
14     {
15         ch = (char)in_stream.get();
16         if (isspace(ch))
17             count++;
18         if (count >= 2){
19             ch = ' ';
20             count = 0;}
21         else
22         {
23             out_stream << ch; }
24     }
```

OUTPUT:

The New file is been created and data updated successfully .

5. A file contains a list of telephone numbers in the following form

John 23456

Ahmed 9976

The names contain only one word and the names and telephone numbers are separated by white spaces.

Write a program to read the file and output the list in two columns. The names should be left-justified and the numbers right-justified.

CODE:

```
Assignment_6 > 5.cpp > Person > put_data()
1  #include <bits/stdc++.h>
2  using namespace std;
3  class Person{
4  public:
5      char name[10];
6      int PhNo;
7      void input_data()
8      {   cout << "Enter the Name:";
9          cin >> name;
10         cout << "Enter the PhNo:";
11         cin >> PhNo;}
12     void put_data(){
13         cout << setw(10) << name << setw(10) << PhNo << endl;
14     };
15
16     int main()
17     {   Person rec;
18         int Phone, pos, choice, offset, i;
19         fstream fp;
20         ifstream in;
21         ofstream out;
22         char nm[20];
23         in.open("test.txt", ios::in | ios::binary);
24         cout << "\nThe contents of file are:\n";
25         while (in.read((char *)&rec, sizeof(rec))){
26             rec.put_data();}
27         in.close();
28         return 0;
29     }
30
```

OUTPUT:

Contents of File are:

Ahmed 3456353456

John 2385939326

6. Write a program that will create a data file containing the First of telephone numbers given in above example- Use a class object to store each set of data.

CODE:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 class Person{
5 public:
6     char name[10];
7     int PhNo;
8
9     void input_data(){
10         cout << "Enter the Name:";
11         cin >> name;
12         cout << "Enter the PhNo:";
13         cin >> PhNo;
14     }
15     void put_data(){
16         cout << setw(10) << name << setw(10) << PhNo << endl; }
17 };
18
19 int main()
20 {
21     Person rec;
22     int Phone, pos, choice, offset, i;
23     fstream fp;
24     ifstream in;
25     ofstream out;
26     char nm[20];
27     rec.input_data();
28     char ch;
29     cin.get(ch);
30     out.open("test.txt", ios::out | ios::app | ios::binary);
31     out.write((char *)&rec, sizeof(rec));
32     out.close();
33     return 0;
34 }
```

OUTPUT:

John 2385939326

Ahmed 7829377458

Mayur 2893050346

7. Write an interactive, menu-driven program that will access the file created above example and implement the following tasks.

- (a) Determine the telephone number of the specified person.
- (b) Determine the name if a telephone number is known.
- (c) Update the telephone number, whenever there is a change

CODE:

```
Assignment_6 > 7.cpp > main()
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  class Person
5  {
6  public:
7      char name[10];
8      int PhNo;
9
10     void input_data()
11     {
12         cout << "Enter the Name:";
13         cin >> name;
14         cout << "Enter the PhNo:";
15         cin >> PhNo;
16     }
17     void put_data()
18     {
19         cout << setw(10) << name << setw(10) << PhNo << endl;
20     }
21 };
22
23 int main()
24 {
25     Person rec;
26     int Phone, pos, choice, offset, i;
27     fstream fp;
28     ifstream in;
29     ofstream out;
30     char nm[20];
31
32     char ans;
```

Assignment_6 > 7.cpp > main()

```
31
32     char ans;
33     do
34     {
35         cout << "\n1.Determine Name if telephone number is specified";
36         cout << "\n2.Determine telephone if name is specifeied";
37         cout << "\n3.Update telephone number";
38         cout << "\nEnter the choice:";
39         cin >> choice;
40         switch (choice)
41         {
42             case 1:
43                 cout << "\nEnter the phone No:";
44                 cin >> Phone;
45                 fp.open("test.txt", ios::ate | ios::in | ios::out | ios::binary);
46                 fp.seekg(0, ios::beg);
47                 pos = -1;
48                 i = 0;
49                 while (fp.read((char *)&rec, sizeof(rec)))
50                 {
51                     if (Phone == rec.PhNo)
52                     {
53                         pos = i;
54                         break;
55                     }
56                     i++;
57                 }
58                 offset = pos * sizeof(rec);
59                 fp.seekp(offset);
60                 fp.read((char *)&rec, sizeof(rec));
61                 cout << "\nName:" << rec.name;
62                 fp.close();
```

```

5.cpp 7.cpp x
assignment_6 > 7.cpp > main()
64
65     case 2:
66         cout << "\nEnter the Name:";
67         cin >> nm;
68         fp.open("test.txt", ios::ate | ios::in | ios::out | ios::binary);
69         fp.seekg(0, ios::beg);
70         pos = -1;
71         i = 0;
72         while (fp.read((char *)&rec, sizeof(rec)))
73         {
74             if ((strcmp(nm, rec.name)) == 0)
75             {
76                 pos = i;
77                 break;
78             }
79             i++;
80         }
81         offset = pos * sizeof(rec);
82         fp.seekp(offset);
83         fp.read((char *)&rec, sizeof(rec));
84         cout << "\nTelephone Number:" << rec.PhNo;
85         fp.close();
86         break;
87
88     case 3:
89         cout << "\nEnter the Name:";
90         cin >> nm;
91         fp.open("test.txt", ios::ate | ios::in | ios::out | ios::binary);
92         fp.seekg(0, ios::beg);
93         pos = -1;
94         i = 0;
95         while (fp.read((char *)&rec, sizeof(rec)))

```

```
5.cpp 7.cpp x
Assignment_6 > 7.cpp > main()
95 while (fp.read((char *)&rec, sizeof(rec)))
96 {
97     if ((strcmp(nm, rec.name)) == 0)
98     {
99         pos = i;
100         break;
101     }
102     i++;
103 }
104 offset = (pos) * sizeof(rec);
105 fp.seekp(offset);
106 cout << "\nCurreent Phone :" << rec.PhNo;
107 cout << "\nEnter new telephoine Number:";
108 cin >> Phone;
109 rec.PhNo = Phone;
110 fp.write((char *)&rec, sizeof(rec)) << flush;
111 cout << "\nrecord updated !!\n";
112 fp.seekg(0);
113 while (fp.read((char *)&rec, sizeof(rec)))
114 {
115     rec.put_data();
116 }
117 fp.close();
118 break;
119 }
120 cout << "\n Do You want to continue?(y/n)";
121 cin >> ans;
122 } while (ans == 'y');
123 return 0;
124 }
```

OUTPUT:

```
1
John: 2385939326
2
2385939326
The Name of the Contact is John
3
John 2345901829
The phone directory is successfully updated.
```