COP4020 Programming Languages C++
Assignment 3

Edelis Molina

November 2022

Contents

1 Problem 1 **3**

2 Problem 2 **3**

3 Problem 3 **3**

4 Problem 4 **4**

5 Problem 5 **5**

6 Problem 6 **6**

7 Problem 7 **7**

8 Problem 8 **7**

9 Problem 9 **8**

10 Problem 10 **9**
10.1 Employee.h 9
10.2 problem10.cpp 10

11 Problem 12 (Instructions skip number 11) **10**
11.1 Computer.h 10
11.2 problem12.cpp 11

12 Problem 13 **12**
12.1 Student.h 12
12.2 problem13.cpp 13

13 Problem 14 **14**
13.1 Employee14.h 14
13.2 problem14.cpp 15

14 Problem 15 **16**
14.1 Stud.h 16
14.2 problem15.cpp 16

15 Problem 16 **18**
15.1 Item.h 18
15.2 problem16.cpp 19

1 Problem 1

```
#include <fstream>

int main()
{
    std::ofstream myFile;
    myFile.open("NOTES.TXT");

    for (int i = 1; i <= 100; i++)
        myFile << i << std::endl;

    myFile.close();

    return 0;
}
```

2 Problem 2

```
#include <fstream>
#include <string>
using namespace std;

int main()
{
    string str = "Time is a great teacher but unfortunately it kills all its pupils. Berlioz

    ofstream fout;
    fout.open("OUT.TXT");
    fout << str;

    fout.close();

    return 0;
}
```

3 Problem 3

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

void readAlphabets();
```

```
int main()
{
    readAlphabets();
    return 0;
}

void readAlphabets()
{
    ifstream fin;
    fin.open("OUT.TXT");

    char ch;
    int cnt = 0;
    while (!fin.eof())
    {
        fin.get(ch);
        if (isalpha(ch))
        {
            cnt++;
        }
    }
    cout << "Number of alphabets present in the file -> " << cnt << endl;
    fin.close();
}
```

4 Problem 4

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

void readNumOfBlanks();

int main()
{
    readNumOfBlanks();
    return 0;
}

void readNumOfBlanks()
{
    ifstream fin;
    fin.open("OUT.TXT");
    char ch;
    int count = 0;
    while (!fin.eof())
    {
        fin.get(ch);
        if (isblank(ch))
        {
            count++;
        }
    }
    cout << "Number of blanks present in the file -> " << count << endl;
    fin.close();
}
```

5 Problem 5

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

void countNumOfWords();

int main()
{
    countNumOfWords();
    return 0;
}

void countNumOfWords()
{
    ifstream fin;
    fin.open("OUT.TXT");

    char ch;
    int words = 0;
    while (!fin.eof())
    {
        fin.get(ch);
        // if it is space, increment word count
        if (ch == ' ')
        {
            words++;
        }
    }
    // at eof, all spaces have been counted. Increment words one more time for correct word
    words++;

    cout << "Number of words present in the file -> " << words << endl;
    fin.close();
}
```

6 Problem 6

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

void countWordThe();

int main()
{
    countWordThe();
    return 0;
}

void countWordThe()
{
    ifstream fin;
    fin.open("STORY.TXT");

    char ch;
    int count = 0;
    char word[100];

    while (!fin.eof())
    {
        // >> operator separates on whitespace
        fin >> word;
        if (strcasecmp(word, "the") == 0) // compare without sensitivity to case
        {
            count++;
        }
    }

    cout << "Number of ’the’ as independent word in the file -> " << count << endl;
    fin.close();
}
```

7 Problem 7

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

void countLines();

int main()
{
    countLines();
    return 0;
}

void countLines()
{
    ifstream fin;
    fin.open("STORY1.TXT");

    string line;
    int count = 0;

    while (!fin.eof())
    {
        getline(fin, line);
        if (line[0] != 'A')
        {
            count++;
        }
    }

    cout << "Number of lines not starting with A are -> " << count << endl;
    fin.close();
}
```

8 Problem 8

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

void copyToUpper();
```

```
int main()
{
    copyToUpper();
    return 0;
}

void copyToUpper()
{
    ifstream fin;
    ofstream fout;

    fin.open("FIRST.TXT");
    fout.open("SECOND.TXT");

    char c;
    while (!fin.eof())
    {
        fin.get(c);
        c = toupper(c);
        fout << c;
    }
    fin.close();
    fout.close();
}
```

9 Problem 9

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

void copyToUpper();

int main()
{
    copyToUpper();
    return 0;
}

void copyToUpper()
{
    ifstream fin;
    ofstream fout;
    fin.open("FIRST1.TXT");
    fout.open("SECOND1.TXT");

    char word[100];
    while (!fin.eof())
    {
        fin >> word;
        if (word[0] == ’a’ || word[0] == ’e’ || word[0] == ’i’ || word[0] == ’o’ || word[0]
            fout << word << " ";
    }
    fin.close();
    fout.close();
}
```

10 Problem 10

10.1 Employee.h

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

class Employee
{
    int ENO;
    char ENAME[10];

public:
    void GETIT()
    {
        cout << "Enter employee name: ";
        // replace gets for getline due to compiler issues
        // gets(ENAME);
        cin.getline(ENAME, 10);
        cout << "Enter employee number: ";
        cin >> ENO;
    }

    void SHOWIT()
    {
        cout << ENO << " " << ENAME << endl;
    }
};
```

10.2 problem10.cpp

```
#include <fstream>
#include <iostream>
#include "Employee.h"

using namespace std;

int main()
{
  fstream myFile("employee.bin", ios::out | ios::in | ios::binary);
  if (!myFile)
  {
    cout << "File could not be opened" << endl;
    return 1;
  }

  Employee employee1, employee2;

  // write employee object to bin file
  employee1.GETIT();
  myFile.write((char *)&employee1, sizeof(Employee));

  // reposition put pointer to start of the file to read Employee object
  myFile.seekp(0);
  myFile.read((char *)&employee2, sizeof(Employee));
  employee2.SHOWIT();

  myFile.close();

  return 0;
}
```

11 Problem 12 (Instructions skip number 11)

11.1 Computer.h

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

class Computer
{
    char chiptype[100];
    int speed;

public:
    void getdetails()
    {
        cout << "Enter chip speed: ";
        cin >> speed;
        // before using getline, flush input buffer
        cin.ignore();
        cout << "Enter chiptype: ";
        // gets(chiptype);
        cin.getline(chiptype, 100);
    }

    void showdetails()
    {
        cout << "Chip: " << chiptype << endl
             << "Speed = " << speed << endl;
    }
};
```

11.2 problem12.cpp

```
#include <iostream>
#include <fstream>
#include "Computer.h"

using namespace std;

int main()
{
  fstream file("CHIP.DAT", ios::out | ios::in | ios::binary);
  if (!file)
  {
    cout << "File could not be opened" << endl;
    return 1;
  }

  // First, write Computer objects to file
  Computer outComputer;
  int count;

  cout << "Enter number of records to write: ";
  cin >> count;

  while (count > 0)
  {
    outComputer.getdetails();
    file.write(reinterpret_cast<char *>(&outComputer), sizeof(Computer));
    count--;
  }

  // Second, read objects from file and display them
  Computer inComputer;
  file.seekp(0);
  while (file.peek() != EOF)
  {
    file.read(reinterpret_cast<char *>(&inComputer), sizeof(Computer));
    inComputer.showdetails();
  }

  file.close();

  return 0;
}
```

12 Problem 13

12.1 Student.h

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

class Student
{
  char S_Admno[10]; // Admission number of student
  char S_Name[30];  // Name of student
  int Percentage;   // Marks Percentage of student

public:
  void EnterData()
  {
    cout << "Enter Percentage 1-100: ";
    cin >> Percentage;
    cin.ignore();
    cout << "Enter Admission number: ";
    cin.getline(S_Admno, 10);
    cout << "Enter Student Name: ";
    cin.getline(S_Name, 30);
  }

  void DisplayData()
  {
    cout << setw(10) << S_Admno;
    cout << setw(30) << S_Name;
    cout << setw(10) << Percentage << endl;
  }

  int ReturnPercentage()
  {
    return Percentage;
  }
};
```

12.2 problem13.cpp

```
#include <iostream>
#include <fstream>
#include "Student.h"

using namespace std;

int main()
{
  fstream file("STUDENT.DAT", ios::out | ios::in | ios::binary);
  if (!file)
  {
    cout << "File could not be opened" << endl;
    return 1;
  }

  // First, write Student objects to file
  Student outStudent;
  int count;

  cout << "Enter number of Student records to write: ";
  cin >> count;

  while (count > 0)
  {
    outStudent.EnterData();
    file.write(reinterpret_cast<char *>(&outStudent), sizeof(Student));
    count--;
  }

  // Second, read objects from file and display them
  Student inStudent;
  file.seekp(0);
  while (file.peek() != EOF)
  {
    file.read(reinterpret_cast<char *>(&inStudent), sizeof(Student));
    if (inStudent.ReturnPercentage() > 75)
    {
      inStudent.DisplayData();
    }
  }

  file.close();

  return 0;
}
```

13 Problem 14

13.1 Employee14.h

```
#include <iostream>
using namespace std;
class Employee14
{
  int Eno;
  char Ename[20];

public:
  // Function to count the total number of records
  int Countrec()
  {
    fstream File;
    File.open("EMP.DAT", ios::binary | ios::in);

    // Move file pointer to EOF and get current position
    File.seekg(0, ios::end);
    int Bytes = File.tellg();
    int Count = Bytes / sizeof(Employee14);
    File.close();
    return Count;
  }

  void readData(int eno, char *ename)
  {
    this->Eno = eno;
    strcpy(this->Ename, ename);
  }
  void displayData()
  {
    cout << "Employee Number: " << Eno << " Employee Name: " << Ename << endl;
  }
};
```

13.2 problem14.cpp

```
#include <iostream>
#include <fstream>
#include "Employee14.h"

using namespace std;

int main()
{
  fstream file("EMP.DAT", ios::out | ios::in | ios::binary);
  if (!file)
  {
    cout << "File could not be opened" << endl;
    return 1;
  }

  // First, write Employee objects to file
  Employee14 outEmployee;
  int count;
  int Eno;
  char Ename[20];

  cout << "Enter number of Employee records to write: ";
  cin >> count;

  while (count > 0)
  {
    cout << "Enter Eno (integer): ";
    cin >> Eno;
    // before using getline, flush input buffer
    cin.ignore();
    cout << "Enter Ename (string): ";
    cin.getline(Ename, 20);

    outEmployee.readData(Eno, Ename);
    file.write(reinterpret_cast<char *>(&outEmployee), sizeof(Employee14));
    count--;
  }
  // Count records
  cout << "Number of Employee Records -> " << outEmployee.Countrec() << endl;

  file.close();

  return 0;
}
```

14 Problem 15

14.1 Stud.h

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

class Stud
{
  int Rno;
  char Name[20];

public:
  void Enter()
  {
    cout << "Enter Student Rno: ";
    cin >> Rno;
    cout << "Enter Student Name: ";
    cin >> Name;
  }

  void Display()
  {
    cout << "Student Rno: " << Rno << " Student Name: " << Name << endl;
  }
};
```

14.2 problem15.cpp

```
#include <iostream>
#include <fstream>
#include "Stud.h"

using namespace std;

int main()
{
  fstream file("STUD.DAT", ios::out | ios::in | ios::binary);
  if (!file)
  {
    cout << "File could not be opened" << endl;
    return 1;
  }

  // First, write Student objects to file
  Stud outStud;
  int count;

  cout << "Enter number of Student records to write: ";
  cin >> count;
  while (count > 0)
  {
    outStud.Enter();
    file.write(reinterpret_cast<char *>(&outStud), sizeof(Stud));
    count--;
  }

  cout << "Would you like to add a new student? (y/n) ";
  char c;
  cin >> c;
  if (c == ’y’ || c == ’Y’)
  {
    outStud.Enter();
    file.write(reinterpret_cast<char *>(&outStud), sizeof(Stud));
  }
  else
    cout << "No new records added to the file" << endl;

   Stud inStud;
  file.seekp(0);

  while (file.peek() != EOF)
  {
    file.read(reinterpret_cast<char *>(&inStud), sizeof(Stud));
    inStud.Display();
  }

  file.close();

  return 0;
}
```

15 Problem 16

15.1 Item.h

```
#include <fstream>
#include <iostream>
#include <string.h>

using namespace std;
class Item
{
private:
  int Ino;
  char Item[20];

public:
  // Function to search and display the content from a particular record number
  void Search(int RecNo)
  {
    fstream File;
    File.open("STOCK.DAT", ios::binary | ios::in);
    if (!File)
    {
      cout << "File could not be opened for writing" << endl;
      return;
    }

    // Move the file pointer to position the file before writing
    File.seekp((RecNo - 2) * sizeof(Item));

    File.read((char *)this, sizeof(Item));
    cout << "Item with RecNo = " << RecNo << endl
         << Ino << " ==> " << Item << endl;
    File.close();
  }

  // Function to modify the content of a particular record number
  void Modify(int RecNo)
  {
    fstream File;
    File.open("STOCK.DAT", ios::binary | ios::in | ios::out);
    if (!File)
    {
      cout << "File could not be opened for reading" << endl;
      return;
    }
    cout << "Modify Item with RecNo = " << RecNo << endl;
    cout << "Enter new Ino: ";
    cin >> Ino;
    cin.ignore();
    cout << "Enter new Item: ";
    cin.getline(Item, 20);

    // Move the file pointer to position the file before reading
    File.seekg((RecNo - 2) * sizeof(Item));

    File.write((char *)this, sizeof(Item));
    File.close();
  }

  // Added function to create Item objects

  void readData(int ino, char *item)
  {
    this->Ino = ino;
    strcpy(this->Item, item);
  }
};
```

15.2 problem16.cpp

```
#include <iostream>
#include <fstream>
#include "Item.h"

using namespace std;

int main()
{
  // First write data to file
  fstream file("STOCK.DAT", ios::out | ios::binary);
  if (!file)
  {
    cout << "File could not be opened" << endl;
    return 1;
  }

  Item outItem;
  int count;
  int Ino;
  char Item[20];
  cout << "Enter number of records to write: ";
  cin >> count;

  while (count > 0)
  {
    cout << "Enter Ino (integer): ";
    cin >> Ino;
    // before using getline, flush input buffer
    cin.ignore();
    cout << "Enter Item (string): ";
    cin.getline(Item, 20);

    outItem.readData(Ino, Item);
    file.write(reinterpret_cast<char *>(&outItem), sizeof(Item));
    count--;
  }

  // Search Item in position 1 and display its content
  outItem.Search(2);

  // Modify Item in position 1
  outItem.Modify(2);
  outItem.Search(2);

  file.close();

  return 0;
}
```