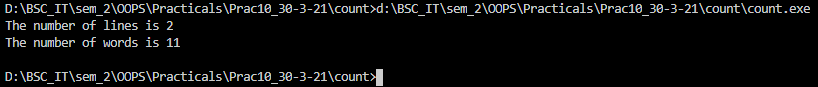
Practical 10: File Handling

1. Aim: Design a class File Demo open a file in read mode and display the total number of words and lines in the file.

Code:

|  |
| --- |
| #include<iostream>  #include<fstream>  #include<sstream>  #include<vector>  #include<string>  #include<algorithm>  using std::cout;  using std::string;  using std::fstream;  using std::ios;  using std::getline;  using std::vector;  using std::stringstream;  int main(){      int lineCount = 0, wordCount = 0;      vector<string> row;      string line, feild;      fstream fileVar("test.txt", ios::in);      while (!fileVar.eof())      {          row.clear();          getline(fileVar, line);          stringstream s(line);          while (getline(s, feild, ' ')){              row.push\_back(feild);          }          wordCount += row.size();          lineCount++;      }      cout << "The number of lines is "<<lineCount<<"\n";      cout << "The number of words is "<<wordCount<<"\n";      return 0;  } |

Output:

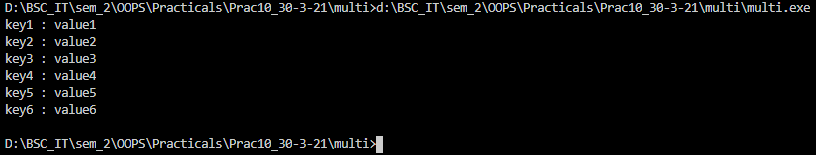


1. Aim: Design a class to handle multiple files and file operations.

Code:

|  |
| --- |
| #include<iostream>  #include<fstream>  using std::cout;  using std::fstream;  using std::ios;  using std::string;  using std::getline;  int main(){      fstream obj1, obj2;      obj1.open("keys.txt", ios::out);      obj2.open("values.txt", ios::out);      string keys[6] = {"key1", "key2", "key3", "key4", "key5", "key6"};      string values[6] = {"value1", "value2", "value3", "value4", "value5", "value6"};      for (int i = 0; i < 6; i++){          obj1 << keys[i] << "\n";          obj2 << values[i] << "\n";      }        obj1.close();      obj2.close();      obj1.open("keys.txt", ios::in|ios::out);      obj2.open("values.txt", ios::in|ios::out);      string temp1, temp2;      int count = 0;      while (count < 6){          getline(obj1, temp1);          getline(obj2, temp2);          cout << temp1 <<" : "<< temp2 <<"\n";          count++;      }      obj1.close();      obj2.close();      return 0;  } |

Output:

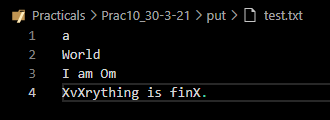


1. Aim: Functions for Manipulation of File Pointers using put pointers.

Code:

|  |
| --- |
| #include<iostream>  #include<fstream>  using std::fstream;  using std::cout;  using std::ios;  int main()  {      fstream obj2;      obj2.open("test.txt",ios::in|ios::out);      cout<<"The first location in the file: "<<obj2.tellp()<<"\n";      char c;      while(obj2)      {          c = obj2.get();          if(c == 'e')          {              obj2.seekp(-1, ios::cur);              obj2.put('X');          }      }      obj2.close();      obj2.open("test.txt",ios::in);      while(obj2.eof())      {          c = obj2.get();          cout << c;      }      obj2.close();      return 0;  } |

Output:

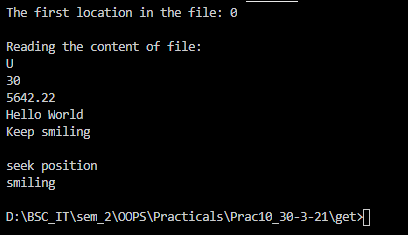


1. Aim: Functions for Manipulation of File Pointers using get pointers.

Code:

|  |
| --- |
| #include<iostream>  #include<fstream>  using std::fstream;  using std::cout;  using std::ios;  int main()  {      fstream obj1;      obj1.open("test.txt", ios::out);      obj1<<"U"<<"\n";      obj1<<"30"<<"\n";      obj1<<"5642.22"<<"\n";      obj1<<"Hello World"<<"\n";      obj1<<"Keep smiling"<<"\n";      obj1.close();      fstream obj2;      obj2.open("test.txt",ios::in);      cout<<"The first location in the file: "<<obj2.tellg()<<"\n";      char ch;      cout<<"\nReading the content of file: \n";      while(obj2)      {          ch=obj2.get();          cout<<ch;      }      obj2.clear();      cout << "\nseek position\n";      obj2.seekg(-9, ios::end);      while(obj2)      {          ch=obj2.get();          cout<<ch;      }      obj2.close();      return 0;  } |

Output:



Write-Up:

