

\* Purpose : Classwork.

\* Date : 05/01/2026

\* Author : Vikas Srivastava

\* ID : 55984

\* Batch ID : 25SUB4505

```
c cmdLineArgs.c
1 // Purpose - Write a program to display command-line arguments in C, showing the count of arguments (argc) and listing each argument (argv).
2
3 #include <stdio.h>
4
5 int main(int argc, char *argv[]){
6     printf("Count of args: %d\n", argc);
7     printf("Argument List: \n");
8     for(int cnt = 0; cnt < argc; cnt++){
9         printf("\t\targv[%d] ==> %s\n", cnt, argv[cnt]);
10    }
11 }
```

Count of args: 1  
Argument List:  
| argv[0] ==> c:\Users\VIKAS SRIVASTAVA\OneDrive\Desktop\C\_CPP\Day\_10\Classwork\cmdLineArgs

```
c fileBinary.cpp
1 // Purpose - Write a program to read a binary file into an object of a class in C++ and display the object's data, demonstrating default initialization and
2 // file input using ifstream.
3
4 #include <iostream>
5 #include <cstring>
6 #include <iostream>
7 using namespace std;
8
9 class Employee{
10     int id;
11     char name[40];
12     double sal;
13 public:
14     Employee(int num=001, const char *na="Sample #1", double sa = 15000.0):id(num), sal(sa){
15         strcpy(name, na);
16     }
17     void disp(){
18         cout<<"Output Num: "<<id<<"\tName: "<<name<<"\tSal: "<<sal<<endl;
19     }
20 };
21
22 int main(){
23     Employee emps;
24
25     emps.disp(); //default values
26
27     ifstream obj("mydata.dat", ios::binary);
28
29     if(obj){
30         obj.read((char *)&emps, sizeof(emps));
31         emps.disp();
32     }
33 }
```

Output Num: 1001 Name: Sample #1 Sal: 15000  
Output Num: 1101 Name: Karthik Sal: 15000.5

```
c fileline.cpp
1 // Purpose - Write a program to read a text file line by line in C++ and display its contents on the console using ifstream and getline().
2
3 #include <iostream>
4 #include <iostream>
5 using namespace std;
6
7
8 int main(){
9     ifstream obj("iothree.cpp");
10    string str;
11    while (getline(obj,str)) //word by word
12        cout<<str<<endl;
13 }
```

```
c fileLineError.cpp X
1 // Purpose - Write a program to open a text file in C++ with error checking, read it line by line, and display its contents on the console.
2
3 #include <iostream>
4 #include <fstream>
5 using namespace std;
6
7 int main(){
8     char fileName[] = "ioThree.txt";
9     ifstream fobj(fileName);
10    string str;
11    if (!fobj){
12        cout<<fileName<<" file NOT FOUND"<<endl;
13        return 1;
14    }
15
16    while (getline(fobj,str)) //word by word
17        cout<<str<<endl;
18
19    }
20 }
```

ioThree.txt file NOT FOUND

```
c fileOpen.cpp X
1 // Purpose - Write a program to write data to a text file in C++ using ofstream.
2
3 #include <iostream>
4 #include <fstream>
5 using namespace std;
6
7 int main(){
8     ofstream fobj("fileOut.txt");
9     fobj<<"Sending data into the file"<<endl;
10
11 }
```

```
c fileOpenTwo.cpp X
1 // Purpose - Write a program to append data to an existing text file in C++ using ofstream with ios::app mode.
2
3 #include <iostream>
4 #include <fstream>
5 using namespace std;
6
7 int main(){
8     ofstream fobj("fileOut.txt", ios::app);
9     fobj<<"Sending data into the file"<<endl;
10
11 }
```

```
c fileOutBinary.cpp X
1 // Purpose - Write a program to write an object of a class to a binary file in C++, demonstrating binary file output using ofstream.
2
3 #include <iostream>
4 #include <fstream>
5 #include <cstring>
6 using namespace std;
7
8 class Employee{
9     int id;
10    char name[40];
11    double sal;
12 public:
13     Employee(int num=1001, const char *na="Sample #1", double sa = 15000.05):id(num), sal(sa){
14         strcpy(name, na);
15     }
16     void disp(){
17         cout<<"Output Num: "<<id<<"\tName: "<<name<<"\tSal: "<<sal<<endl;
18     }
19 };
20
21 int main(){
22     Employee emps = Employee(1101,"Karthik", 15000.45);
23
24     ofstream fobj("myData.dat", ios::binary);
25
26     fobj.write((char *) &emps, sizeof(emps));
27     emps.disp();
28
29 }
```

Output Num: 1101 Name: Karthik Sal: 15000.5

```
filePutPointer.cpp X
1 // Purpose - Write a program to demonstrate the use of tellp() in C++, showing how to get the current write position in an output file stream.
2
3 #include <iostream>
4 #include <fstream>
5 using namespace std;
6
7 int main() {
8     ofstream fout("dataTest.txt");
9
10    cout << "Initial position: " << fout.tellp() << endl;
11
12    fout << "Hello";
13    cout << "After writing 'Hello': " << fout.tellp() << endl;
14
15    fout << "World";
16    cout << "After writing ' World': " << fout.tellp() << endl;
17
18    fout.close();
19 }
20 }
```

```
Initial position: 0
After writing 'Hello': 5
After writing ' World': 11
```

```
fileWord.cpp X
1 // Purpose - Write a program to read a text file word by word in C++ using ifstream and the extraction operator (>>) and display the words on the console.
2
3 #include <iostream>
4 #include <fstream>
5 using namespace std;
6
7 int main(){
8     ifstream fobj("ioThree.cpp");
9     string str;
10    while (fobj>>str) //word by word
11        | cout<<str<<" ";
12    }
13
14 }
```

```
ioOne.cpp X
1 // Purpose - Write a program to read input character by character from the standard input in C++ and immediately echo it to the console.
2
3 #include <iostream>
4 using namespace std;
5
6 int main(){
7     char ch;
8     while ( cin >> ch)
9         | cout<<ch;
10    }
11
12 }
```

```
ioTwo.cpp X
1 // Purpose - Write a program to read a text file character by character in C++ using ifstream::get() and display its contents on the console.
2
3 #include <iostream>
4 #include <fstream>
5 using namespace std;
6
7 int main(){
8     ifstream fobj("ioThree.cpp");
9     char ch;
10    while ( fobj.get(ch))
11        cout<<ch;
12    }
13
14 }
```

```
myCat.cpp X
1 // Purpose - Write a program to display the contents of a text file in C++, optionally numbering each line when a -n command-line option is provided.
2
3 #include <iostream>
4 #include <fstream>
5 using namespace std;
6
7 void fileOut(istream& in, ostream &out, bool lineNu=false);
8
9 int main(int argc, char *argv[]){
10    if (argc == 2){
11        ifstream fobj(argv[1]);
12        fileOut(fobj, cout);
13    }else if ((argc == 3) && (argv[1][0] == '-' ) && (argv[1][1]== 'n')){
14        ifstream fobj(argv[2]);
15        fileOut(fobj, cout, true);
16    }
17 }
18
19 void fileOut(istream &in, ostream &out, bool lineNu){
20    char ch;
21    int num = 1;
22    if(lineNu)
23        cout<<num<<"\t";
24    while(in.get(ch)){
25        out<<ch;
26        if (((ch=='\n') && lineNu)
27            | cout<<(num+1)<<"\t";
28    }
29 }
```