312304 - OOP Using C++ (Sem III)

As per MSBTE's K Scheme

CO / CM / IF

Unit 5:File Operation

Marks - 10

Sr.No	Questions	Marks	Year
1.	**List C++ stream classes along with their function.(Any two classes)**	2	S-24
	C++ stream classes: 1. istream class: This class is responsible for handling input stream. It provides number of functions for handling chars, strings and objects such as get, getline, read etc. 2. ios class: The ios class is responsible for providing all input and output facilities to all other stream classes. 3. ostream class: This class is responsible for handling output stream. It provides number of functions for handling chars, strings and objects such as write, put etc. 4. iostream class: This class is responsible for handling both input and output stream as both istream class and ostream class is inherited into it. It provides function of both istream class and ostream class for handling chars, strings and objects such as get, getline, read, put, write etc. 5. streambuf class: This class provide an interface to physical devices through buffer. It acts as a base for filebuf class used for ios files.		
2.	**Give the syntax of fclose() method**	2	S-24
	Syntax: int fclose(FILE* stream);		
3.	**Write a program to count number of lines in a file.**	4	S-24
	#include<iostream> #include<fstream>		

	using namespace std;
	int main()
	{
	int count = 0;
	string line;
	ifstream file("abc.txt");
	while (getline(file, line))
	count++;
	cout << "Numbers of lines in the file : " << count << endl;
	return 0;
	}		

4.	**Develop a C++ program to read content of file abc.txt**	**4**	**S-24**

	#include <iostream>
	#include <fstream>
	using namespace std;
	int main() {
	ifstream inputFile("abc.txt"); // Check if the file was opened successfully
	if (inputFile.is_open()) {
	string line; // Read the file line by line
	while (getline(inputFile, line))
	{
	// Print each line to the console
	cout << line << endl;
	}
	inputFile.close();
	}
	else {
	//cerr << "Error: Could not open file 'abc.txt'." << endl;
	cout << "Error:'abc.txt' File not found." << endl;
	}
	return 0;
	}		

5.	**Give the syntax and use of fclose() fuction.**	**2**	**W-23**
	int fclose(FILE *stream);		
6.	**Write a C++ program for write into a file using file operations.**	**4**	**W-23**
	#include <iostream>		

	#include <fstream>
	#include <string>
	using namespace std;
	int main() {
	ofstream outputFile("data.txt"); // Create an output file stream
	if (!outputFile) {
	cerr << "Error opening file!" << endl;
	return 1;
	}
	string data = "This is some data to be written to the file.";
	outputFile << data << endl; // Write data to the file
	outputFile.close(); // Close the file
	cout << "Data written to data.txt successfully!" << endl;
	return 0;
	}		

7.	**Write a C++ program to copy data from one file to another**	**4**	**W-23**

	#include <iostream>
	#include <fstream>
	using namespace std;
	int main() {
	ifstream inputFile("source.txt");
	ofstream outputFile("destination.txt");
	if (!inputFile) {		

	cerr << "Error opening source file!" << endl;
	return 1;
	}
	if (!outputFile) {
	cerr << "Error opening destination file!" << endl;
	return 1;
	}
	char ch;
	while (inputFile.get(ch)) {
	outputFile << ch;
	}
	inputFile.close();
	outputFile.close();
	cout << "Data copied from source.txt to destination.txt successfully!" << endl;
	return 0;
	}		

8.	**Define file with it's operations.**	**2**	**S-23**
	What is a File? A file is a named collection of data stored in a computer's secondary storage. It's a fundamental unit for storing and organizing information. Files can contain various types of data, such as text, images, audio, video, and executable programs. **File Operations**		

	The primary operations performed on files are: **Creating a file:** This involves allocating space on the storage device and assigning a name to the file. **Opening a file:** Establishes a connection between the program and the file, allowing data to be read from or written to it. Different modes can be specified (e.g., read, write, append). **Reading from a file:** Retrieves data from the file and stores it in memory. **Writing to a file:** Stores data from memory into the file. **Closing a file:** Terminates the connection between the program and the file, releasing system resources. **Seeking:** Moves the file pointer to a specific position within the file. **Deleting a file:** Removes the file from the storage device. **Renaming a file:** Changes the name of the file.		
9.	**Write a program for get and put functions.**	**4**	**S-23**
	get() and put() are member functions of the istream and ostream classes respectively, which are used for input and output operations. They are typically used for character-based input and output. get() function Syntax: istream_object.get(char_variable) Description: Reads a single character from the input stream and stores it in the specified character variable. Return value: Returns the input stream object itself, allowing for chaining. put() function Syntax: ostream_object.put(char_value)		

	Description: Writes a single character to the output stream. Return value: Returns the output stream object itself, allowing for chaining.		
10.	**Write a program for closing a file.**	**4**	**S-23**
	Give syntax and use of fclose ( ) function.(Repeat)	**2**	**W-19**

	#include <iostream>
	#include <fstream>
	using namespace std;
	int main() {
	ofstream outputFile("example.txt");
	// Write some data to the file (optional)
	outputFile << "This is some data to be written to the file." << endl;
	// Close the file
	outputFile.close();
	cout << "File closed successfully." << endl;
	return 0;
	}		

11.	**Explain file modes used to perform file operations.**	**2**	**W-22**
	Write the use of ios : : in and ios : : out.(Repeat)	**2**	**S-19**
	Describe meaning of following: (Repeat) **(i) ios : : in** **(ii) ios : : out**	**2**	**W-18**
	File Modes in C++ File modes determine how a file is opened when using file streams. They specify the type of operations that can be performed on the file.		

	Here are the common file modes in C++: Basic Modes ios::in: Opens a file for input operations. ios::out: Opens a file for output operations. If the file doesn't exist, it's created. ios::app: Opens a file for appending data at the end. If the file doesn't exist, it's created. ios::binary: Opens a file in binary mode.		
12	**List all stream classes used in stream operation.**	**2**	**W-22**
	C++ Stream Classes C++ provides a set of stream classes for handling input and output operations. These classes are organized in a hierarchy with ios as the base class. **Core Stream Classes** ios: This is the base class for all stream classes. It provides fundamental functionalities for input and output operations. istream: Derived from ios, it represents an input stream. It provides functions for reading data from various sources, such as the keyboard or a file. cin: Standard input stream (usually the keyboard) ostream: Derived from ios, it represents an output stream. It provides functions for writing data to various destinations, such as the console or a file. cout: Standard output stream (usually the console) cerr: Standard error stream (usually the console) clog: Standard logging stream (usually the console) iostream: Derived from both istream and ostream, it supports both		

	input and output operations. **File Stream Classes** ifstream: Derived from istream, it represents an input file stream. It's used for reading data from files. ofstream: Derived from ostream, it represents an output file stream. It's used for writing data to files. fstream: Derived from iostream, it supports both input and output operations on files. **Other Stream Classes** streambuf: Provides an interface to the underlying buffer for stream operations. wistream, wostream, wiostream: Wide character versions of istream, ostream, and iostream for handling Unicode characters.		
13.	**Develop c++ program to open and read content of file also write "object oriented" string in file and close it.**	**4**	**W-22**

	#include <iostream>
	#include <fstream>
	#include <string>
	using namespace std;
	int main() {
	string line;
	ifstream inputFile("file.txt"); // Open file for reading
	if (inputFile.is_open()) {
	cout << "File content:" << endl;
	while (getline(inputFile, line)) {		

	cout << line << endl;
	}
	inputFile.close();
	} else {
	cout << "Error opening file!" << endl;
	return 1;
	}
	ofstream outputFile("file.txt", ios::app); // Open file for appending
	if (outputFile.is_open()) {
	outputFile << "\nobject oriented";
	outputFile.close();
	cout << "String 'object oriented' appended to file successfully." << endl;
	} else {
	cout << "Error opening file for appending!" << endl;
	return 1;
	}
	return 0;
	}		

14.	**Develop c++ program to check Detection of end of file.**	**4**	**W-22**
	#include <iostream> #include <fstream> using namespace std;		

	int main() {
	ifstream inputFile("file.txt");
	if (!inputFile) {
	cerr << "Error opening file!" << endl;
	return 1;
	}
	char ch;
	while (inputFile.get(ch)) {
	cout << ch;
	}
	if (inputFile.eof()) {
	cout << "\nEnd of file reached." << endl;
	} else {
	cout << "\nError reading file." << endl;
	}
	inputFile.close();
	return 0;
	}		

15	**List c++ stream classes along with their function. (any two classes)**	**2**	**S-22**
	C++ Stream Classes **1. istream class** Purpose: Handles input operations from various sources like keyboard, files, etc. **Functions:** get(): Reads a single character.		

	getline(): Reads a line of text. ignore(): Skips characters in the input stream. peek(): Looks at the next character without extracting it. operator>>: Extracts formatted data from the input stream. **2. ostream class** Purpose: Handles output operations to various destinations like console, files, etc. **Functions:** put(): Writes a single character to the output stream. write(): Writes a block of characters to the output stream. operator<<: Inserts formatted data into the output stream. flush(): Flushes the output buffer.		
16.	**Explain ios :: app and ios :: in flags**	**2**	**S-22**
	ios::app Purpose: Opens a file for appending data. Behavior: When a file is opened with this flag, any data written to the file is added to the end of the existing content. If the file doesn't exist, it's created. **ios::in** Purpose: Opens a file for input operations. Behavior: Allows reading data from an existing file. The file pointer is positioned at the beginning of the file.		
17.	**Write a C++ program to copy the contents of a source file student 1.txt to a destination file student 2.txt using file operations.**	**4**	**S-22**

	#include <iostream>
	#include <fstream>
	using namespace std;
	int main() {
	ifstream sourceFile("student1.txt");
	ofstream destinationFile("student2.txt");
	if (!sourceFile) {		

	cerr << "Error opening source file!" << endl;
	return 1;
	}
	if (!destinationFile) {
	cerr << "Error opening destination file!" << endl;
	return 1;
	}
	char ch;
	while (sourceFile.get(ch)) {
	destinationFile << ch;
	}
	sourceFile.close();
	destinationFile.close();
	cout << "File copied successfully!" << endl;
	return 0;
	}		

18.	**Write a program to count the number of lines in file.**	**4**	**W-19**

	#include <iostream>
	#include <fstream>
	#include <string>
	using namespace std;
	int main() {
	ifstream inputFile("file.txt");		

	string line;
	int count = 0;
	if (inputFile.is_open()) {
	while (getline(inputFile, line)) {
	count++;
	}
	inputFile.close();
	cout << "Number of lines: " << count << endl;
	} else {
	cout << "Error opening file!" << endl;
	}
	return 0;
	}		

19.	**Write a program that copies contents of one file into another file.**	**6**	**W-19**
	#include <iostream> #include <fstream> using namespace std; int main() { ifstream sourceFile; ofstream destinationFile; char ch; string sourceFileName, destinationFileName; cout << "Enter the source file name: ";		

	cin >> sourceFileName;
	cout << "Enter the destination file name: ";
	cin >> destinationFileName;
	sourceFile.open(sourceFileName, ios::in);
	if (!sourceFile) {
	cerr << "Error opening source file." << endl;
	return 1;
	}
	destinationFile.open(destinationFileName, ios::out);
	if (!destinationFile) {
	cerr << "Error creating destination file." << endl;
	sourceFile.close();
	return 1;
	}
	while (sourceFile.get(ch)) {
	destinationFile << ch;
	}
	sourceFile.close();
	destinationFile.close();
	cout << "File copied successfully." << endl;
	return 0;
	}		

20.	**Write a C++ program to count number of spaces in text file**	**4**	**S-19**
	Write a C++ program to count number of spaces present in contents of	**4**	**W-18**

	file.		

	#include <iostream>
	#include <fstream>
	using namespace std;
	int main() {
	ifstream inputFile("file.txt");
	char ch;
	int spaceCount = 0;
	if (inputFile.is_open()) {
	while (inputFile.get(ch)) {
	if (ch == ' ') {
	spaceCount++;
	}
	}
	inputFile.close();
	cout << "Number of spaces: " << spaceCount << endl;
	} else {
	cout << "Error opening file!" << endl;
	}
	return 0;
	}		

21.	**Write a C++ program to append data from abc .txt to xyz .txt file.**	**6**	**S-19**
	#include <iostream> #include <fstream>		

	using namespace std;
	int main() {
	ifstream sourceFile("abc.txt");
	ofstream destinationFile("xyz.txt", ios::app);
	if (!sourceFile) {
	cerr << "Error opening source file!" << endl;
	return 1;
	}
	if (!destinationFile) {
	cerr << "Error opening destination file!" << endl;
	return 1;
	}
	char ch;
	while (sourceFile.get(ch)) {
	destinationFile << ch;
	}
	sourceFile.close();
	destinationFile.close();
	cout << "Data appended successfully!" << endl;
	return 0;
	}		

22.	**Write a C++ program to write 'Welcome to poly' in a file. Then read the data from file and display it on screen.**	**6**	**W-18**
	#include <iostream> #include <fstream>		

```
#include <string>
using namespace std;
int main() {
    // Write to file
    ofstream outputFile("welcome.txt");
    if (outputFile.is_open()) {
        outputFile << "Welcome to poly";
        outputFile.close();
        cout << "Data written to file successfully." << endl;
    } else {
        cout << "Error opening file for writing!" << endl;
        return 1;
    }
    // Read from file
    ifstream inputFile("welcome.txt");
    string data;
    if (inputFile.is_open()) {
        getline(inputFile, data);
        inputFile.close();
        cout << "Data from file: " << data << endl;
    } else {
        cout << "Error opening file for reading!" << endl;
        return 1;
    }
```

	return 0; }		

Thank You

https://shikshamentor.com/object-oriented-programming-using-c-for-msbte-3k-scheme/

Visit

https://shikshamentor.com/