

An illustration of a collaborative workspace from a top-down perspective. The background is a teal color. In the center, the text "Information Storage & Management" is written in a large, white, sans-serif font. Below it, the text "Lab 6" is written in a smaller, red, sans-serif font. Surrounding the text are various icons and illustrations of hands interacting with technology and documents. At the top left, hands are writing in a spiral notebook. At the top right, a hand points at a laptop screen while another hand uses a mouse. In the middle left, a hand points at a tablet displaying a bar chart. In the middle right, a hand uses a red pen to mark a document. At the bottom left, hands are typing on a laptop. At the bottom center, hands are holding a document with a bar chart. At the bottom right, hands are typing on a laptop. Other items include a coffee cup, a smartphone, a USB drive, and several small notepads and pencils scattered around the workspace.

Information Storage & Management

"Lab 6"

Agenda



- ❑ Read & Write functions
- ❑ Fixed length for organizing files

The write and read Member Functions



The file stream object's used two member function to read and write data to binary file.

- Write member function used to write binary data to a file. The general format of the write function is:

`fileObject.write(address, size);`

- Read member function is used to read binary data from a file into memory.

The general format of the read member function is:

`fileObject.read(address, size);`

Text file & Binary file



Text File	Binary File
Data stored in the files formatted as ASCII text. (Data when stored in a file with the << operator, is converted to text when we double click to open them.)	Data stored in the files in the same form of representation in main memory.
Consists of readable characters separated into lines by newline characters.	Do not normally use anything to separate the data into lines.

Why used Binary Files?

When data is stored in a file with binary format the process of Reading & Writing being faster because, No time is lost in converting the data from one format to another format.

Writing

```
#include<iostream>
#include<fstream>
using namespace std;
struct student
{
    char name[20],
    address[20];
};
```

```
ofstream file;
file.open("aya1.txt", ios::binary);
student stu;
strcpy_s(stu.name, "bfcai");
strcpy_s(stu.address, "Benha");
//cin >> stu.name;
//cin >> stu.address;
////////////////////////////////////
file.write(stu.name,20);
file.write(stu.address,20);
//file.write((char*)&stu, sizeof(stu));
file.close();
```

Reading



```
ifstream file;  
file.open("aya1.txt",ios::binary);  
student stu2;  
file.read((char*)&stu2, sizeof(stu2));  
cout << stu2.name << " " << stu2.address;  
file.close();  
system("pause");
```



THANKS!