**PF Lab no. 3**

**University of Central Punjab**

**Faculty of Information Technology**

**Spring 2023 – 21/Mar/2023**

**Lecturer: Hafiz Bilal Shahid**

**Name: Muhammad Ahmad**

**Roll No.: L1F22BSCS0634**

**Section: B12**

* **Task no. 1: -**

**File making, storing data in it, input and output of information:**

**#include<iostream>**

**#include<fstream>**

**using namespace std;**

**int main()**

**{**

**//declaring char array to store information**

**char inform[50];**

**//creating and writing info on test file**

**ofstream lab;**

**lab.open("test.txt");**

**cout<<"writing to a text file:"<<endl;**

**//taking input of name from user**

**cout<<"Enter your name: ";**

**cin.getline(inform,50);**

**//storing name of user in file**

**lab<<inform<<endl;**

**//taking input from user of age**

**cout<<"Enter your age: ";**

**cin>>inform;**

**//storing in test file**

**lab<<inform<<endl;**

**//closing file**

**lab.close();**

**cout<<"Reading from a text file: ";**

**//now using ifstream to output info**

**ifstream print;**

**print.open("test.txt");**

**if(print.is\_open()){**

**print>>inform;**

**cout<<inform<<endl;**

**print>>inform;**

**cout<<inform<<endl;**

**print.close();**

**}**

**else{**

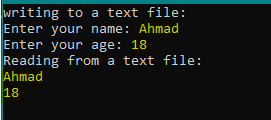
**cout<<”File is not found!”<<endl;**

**}**

**return 0;**

**}**

**Output:**

****

* **Task no. 2: -**

**Marks Problem:**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main() {**

**//declaration and initialization**

**float max\_marks=0, marks;**

**string name, position\_holder;**

**//opening already present file of marks**

**ifstream file1("marks.txt");**

**//creating new file for position holder**

**ofstream file2("position.txt");**

**//loop to find the max marks in class**

**for(int a=0;a<20;a++){**

**file1>>name>>marks;**

**if(marks>max\_marks){**

**position\_holder=name;**

**max\_marks=marks;**

**}**

**}**

**//storing information of position holder in another txt file**

**file2<<position\_holder<<max\_marks;**

**//display of result**

**cout<<position\_holder<<" "<<max\_marks<<endl;**

**//closing of files**

**file1.close();**

**file2.close();**

**return 0;**

**}**

**Output:**

* **Task no. 3: -**

**Admission Form Problem: -**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main() {**

**cout<<"\t\t--Admission Data--"<<endl;**

**//declaration of variables**

**string name,program;**

**int age;**

**char inp='y';**

**//creating txt file to store data of students**

**ofstream file("data.txt");**

**//loop to repeat the input of students data**

**while(inp=='y'||inp=='Y'){**

**//taking input from user**

**cout<<endl<<"Enter Name of the Student: ";**

**cin>>name;**

**cout<<"Enter Age of the Student: ";**

**cin>>age;**

**cout<<"Enter Program of the Student: ";**

**cin>>program;**

**//storing that data in txt file**

**file<<name<<", "<<age<<", "<<program<<endl;**

**//Repeat option**

**cout<<"Do you want to repeat? (y/n)";**

**cin>>inp;**

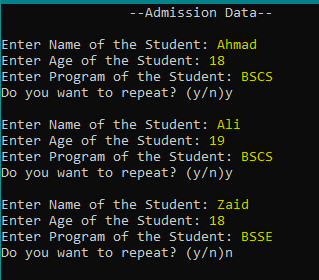
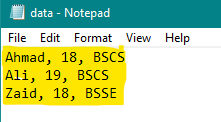
**cin.ignore();**

**}**

**file.close();**

**return 0;**

**}**

**Output: -**

* **Task no. 4: -**

**Grade Analysis Problem: -**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main() {**

**cout << "\t\t--Grade Analysis--" << endl;**

**// Declaration of variables**

**int count[5], total\_count = 0;**

**char grade[5], max\_grade, min\_grade, avg\_grade;**

**float average\_count = 0;**

**// Opening txt file to read data**

**ifstream file("grades.txt");**

**// Reading data from file**

**for (int a = 0; a < 5; a++) {**

**file >> grade[a] >> count[a];**

**total\_count += count[a];**

**}**

**// Finding the grade with the maximum count**

**int max\_count = count[0];**

**max\_grade = grade[0];**

**for (int b = 1; b < 5; b++) {**

**if (count[b] > max\_count) {**

**max\_count = count[b];**

**max\_grade = grade[b];**

**}**

**}**

**cout << "Grade with maximum count: " << max\_grade << “ ”<<max\_count << endl;**

**// Finding the grade with the minimum count**

**int min\_count = count[0];**

**min\_grade = grade[0];**

**for (int c = 1; c < 5; c++) {**

**if (count[c] < min\_count) {**

**min\_count = count[c];**

**min\_grade = grade[c];**

**}**

**}**

**cout << "Grade with minimum count: " << min\_grade << “ ”<<min\_count << endl;**

**// Finding the grade with the average count**

**for (int d = 0; d < 5; d++) {**

**if (count[d] \* 1.0 / total\_count > average\_count) {**

**average\_count = count[d] \* 1.0 / total\_count;**

**avg\_grade = grade[d];**

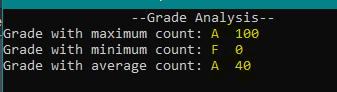
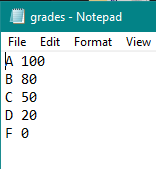
**}**

**}**

**cout << "Grade with average count: " << avg\_grade <<” “<< average\_count \* 100 << endl;**

**return 0;**

**}**

**Output: -**