**1 ) Constructor overloading and destructor:**

**#include <iostream>**

**using namespace std;**

**class Person**

**{**

**private:**

**int age;**

**int salary;**

**public:**

**Person(){**

**age=20;**

**salary =55000;**

**}**

**Person(int a,int b){**

**age=a;**

**salary=b;**

**}**

**int getage(){**

**return age;**

**}**

**int getsalary(){**

**return salary;**

**}**

**~Person(){**

**cout<<"Destructor called in program"<<endl;**

**}**

**};**

**int main()**

**{**

**Person Person1,Person2(43,65000);**

**cout<<"Persion1 Age : "<<Person1.getage()<<endl;**

**cout<<"Persion1 salary : "<<Person1.getsalary()<<endl;**

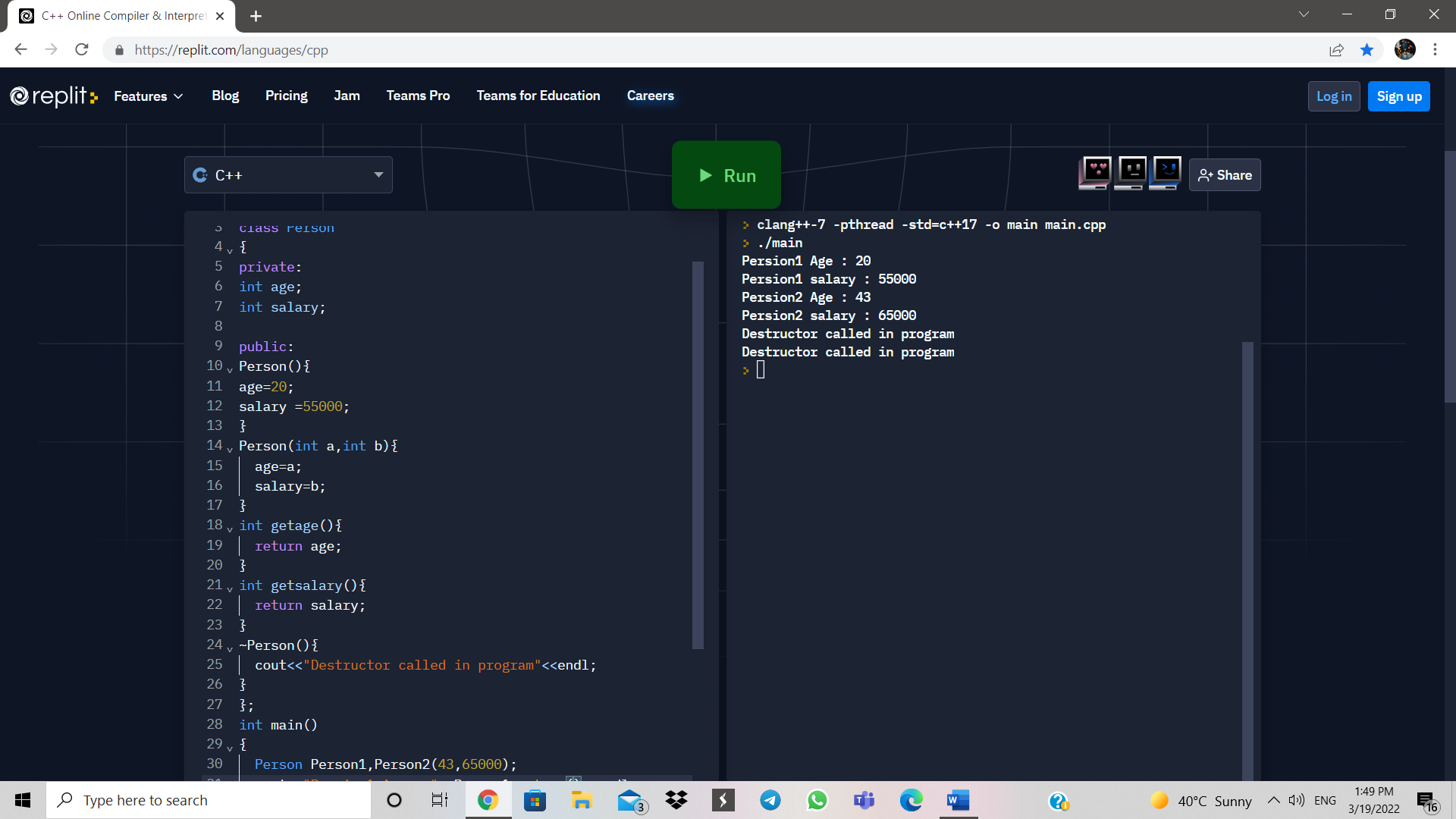
**cout<<"Persion2 Age : "<<Person2.getage()<<endl;**

**cout<<"Persion2 salary : "<<Person2.getsalary()<<endl;**

**return 0;**

**}**

**OUTPUT:**



**4 ) Inheritance :**

**#include <iostream>**

**using namespace std;**

**class Animal**

**{**

**public:**

**void eat(){**

**cout<<"I can eat carrots!"<<endl;**

**}**

**void sleep(){**

**cout<<" I can sleep !"<<endl;**

**}**

**};**

**class Rabit : public Animal**

**{**

**public:**

**void teeth(){**

**cout<<"I have a sharp tooth !"<<endl;**

**}**

**};**

**int main()**

**{**

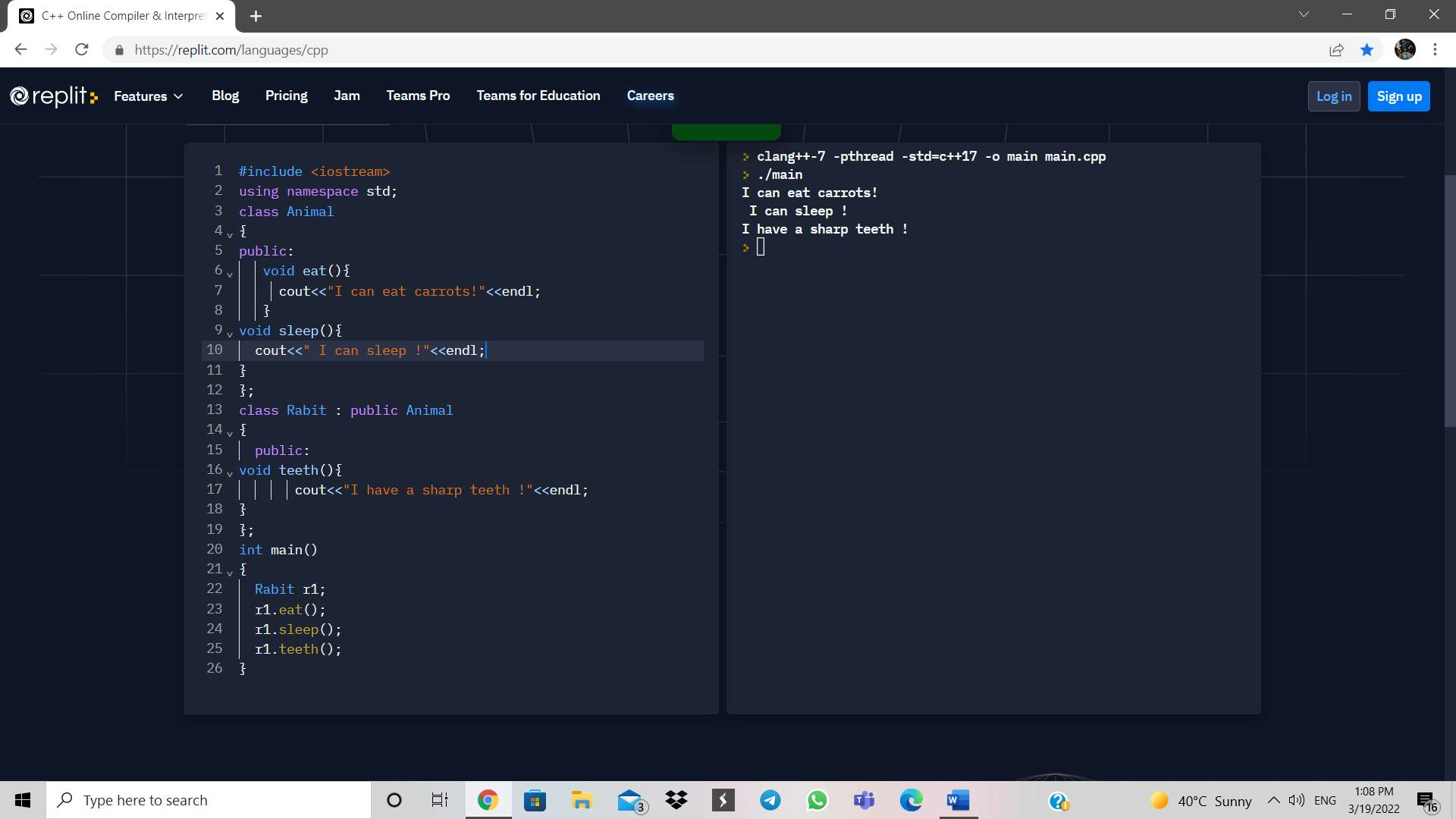
**Rabit r1;**

**r1.eat();**

**r1.sleep();**

**r1.teeth();**

**}**



**6 ) Friend function:**

**#include <iostream>**

**using namespace std;**

**class Box**

**{**

**private:**

**int length;**

**public:**

**Box():length(0){}**

**friend int printLength(Box);**

**};**

**int printLength(Box b)**

**{**

**b.length+=10;**

**return b.length;**

**}**

**int main()**

**{**

**Box b;**

**cout<<"Length of Box : "<<printLength(b)<<endl;**

**return 0;**

**}**

**9 )File Handling:**

**#include <iostream>**

**#include<fstream>**

**using namespace std;**

**struct line**

**{**

**char s[80];**

**};**

**int main()**

**{**

**ofstream fp;**

**fp.open("abc.txt" ,ios::out | ios::binary);**

**line h;**

**int n;**

**n = 1;**

**cout<<"Enter 5 lines : \n";**

**while(n <= 5)**

**{**

**cin.getline(h.s,80);**

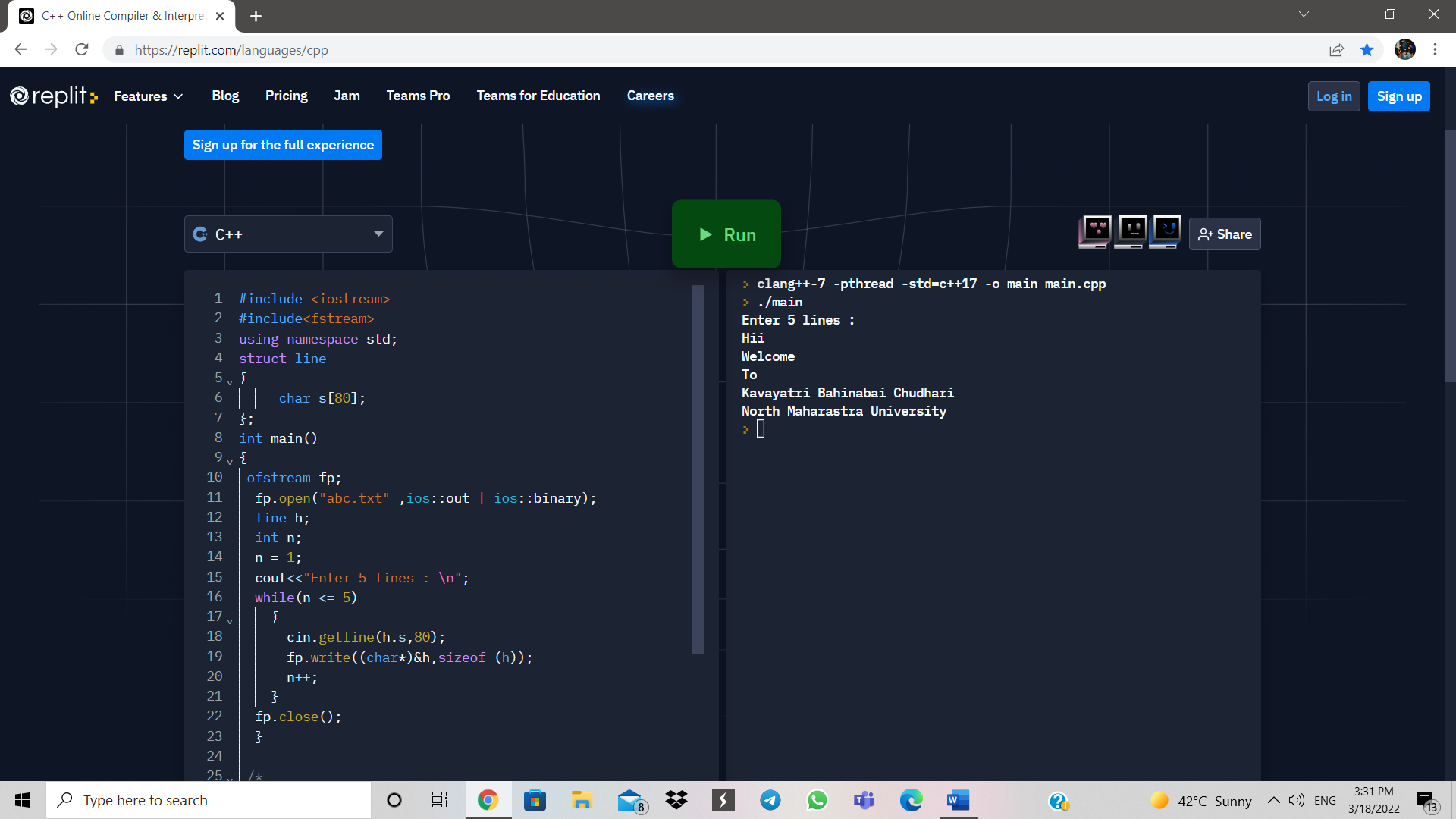
**fp.write((char\*)&h,sizeof (h));**

**n++;**

**}**

**fp.close();**

**}**



**#include <iostream>**

**#include<fstream>**

**using namespace std;**

**struct line**

**{**

**char s[80];**

**};**

**int main()**

**{**

**ofstream ofile;**

**ifstream in;**

**line h;**

**ofile.open("abc.txt",ios::in | ios::binary);**

**int n=0;**

**while(n<5)**

**{**

**in.read((char\*)&h,sizeof (h));**

**cout<<h.s<<endl;**

**n++;**

**}**

**in.close();**

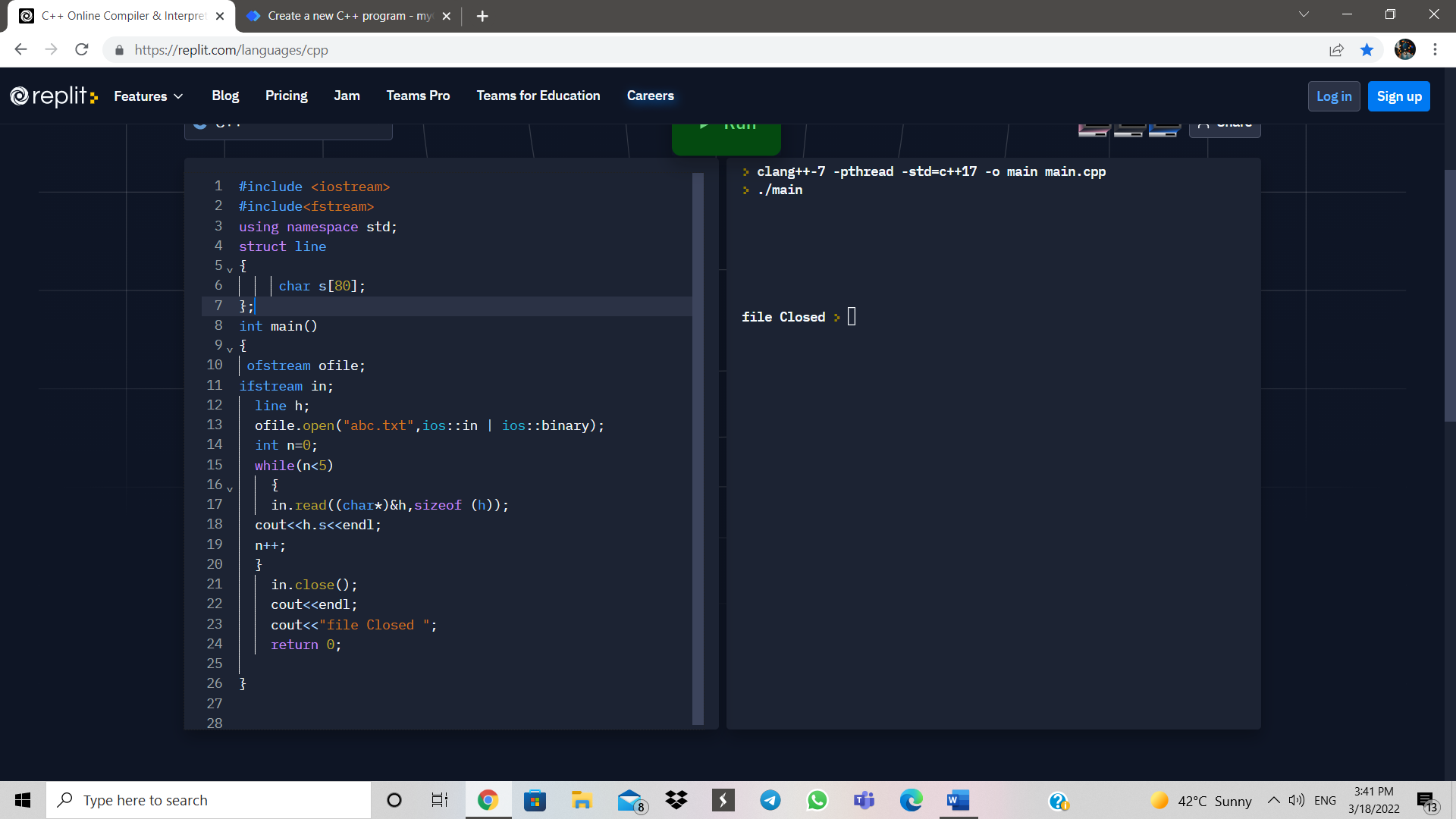
**cout<<endl;**

**cout<<"file Closed ";**

**return 0;**

**}**

**OUTPUT:**



**11 ) Class Templates:**

**#include <iostream>**

**#include<string>**

**using namespace std;**

**template<class t>**

**t minimum(t a,t b)**

**{**

**if (a < b)**

**return a;**

**return b;**

**}**

**int main()**

**{**

**int a=10,b=20;**

**double x=9.9,y=3.5;**

**string p="YUKTA",q="JIMIN";**

**cout<<minimum(a,b)<<endl;**

**cout<<minimum(x,y)<<endl;**

**cout<<minimum(p,q)<<endl;**

**return 0;**

**}**

**OUTPUT:**

