**Write**

|  |  |
| --- | --- |
| #include<iostream>  #include<fstream>  using namespace std;  int main()  {  ofstream fout;  fout.open("text.txt");  fout<<"Welcome";  fout.close();  return 0;  }  OP: Welcome | #include<iostream>  #include<fstream>  using namespace std;  int main()  {  ofstream fout;  fout.open("text.txt", ios::app);  fout<<"GDB";  fout.close();  return 0;  }  OP: WelcomeGDB |
| **Read**  #include<iostream>  #include<fstream>  using namespace std;  int main()  {  ifstream fin;  char ch;  fin.open("text.txt");  while(!fin.eof())  {f  ch=fin.get();  cout<<ch;  }  fin.close();  return 0;  } |  |

|  |  |
| --- | --- |
| #include<iostream>  #include<fstream>  using namespace std;  int main()  {  ifstream fin;  char ch;  fin.open("text.txt");  int position;  position= fin.tellg();  cout<<position;  fin.close();  return 0;  }  o/p: 0 | #include<iostream>  #include<fstream>  using namespace std;  int main()  {  ifstream fin;  char ch;  fin.open("text.txt");  int position;  position= fin.tellg();  cout<<position;  fin>>ch;  position= fin.tellg();  cout<<position;  fin.close();  return 0;  }  o/p: 01 |

Append + tellp() text file already hi hello lika ase

|  |  |
| --- | --- |
| #include<iostream>  #include<fstream>  using namespace std;  int main()  {  ofstream fout;  char ch;  fout.open("text.txt", ios::app);  int position;  position=fout.tellp();  cout<<position;  fout.close();  return 0;  }  O/P: 0 | #include<iostream>  #include<fstream>  using namespace std;  int main()  {  ofstream fout;  char ch;  fout.open("text.txt", ios::app);  int position;  position=fout.tellp();  cout<<position;  fout<<"Welcome";  position=fout.tellp();  cout<<position;  fout.close();  return 0;  }  O/P: 0 7 |

|  |  |
| --- | --- |
| Seekg()  #include<stdio.h>  #include<iostream>  #include<fstream>  using namespace std;  int main()  {  ifstream fin;  fin.open("test.txt"); // HI hello  char ch;  int position;  position=fin.tellg();  cout<<position; //0  ch=fin.get();  cout<<ch; //H  position=fin.tellg();  cout<<position; //1  cout<<(char)fin.get(); //I //cast character  fin.seekg(0); //abar shuru te niye jabe  cout<<"\n"<<fin.tellg(); //0  cout<<(char)fin.get(); //H  cout<<(char)fin.get(); //I  fin.seekg(4);  cout<<"\n"<<fin.tellg(); //4  cout<<(char)fin.get(); //e  fin.seekg(2,ios\_base::cur);  cout<<"\n"<<fin.tellg(); //7  cout<<(char)fin.get(); //o  fin.seekg(-2,ios\_base::cur);  cout<<"\n"<<fin.tellg(); //6, cursor chilo eof()  cout<<(char)fin.get(); //l  fin.seekg(-1,ios\_base::end);  cout<<"\n"<<fin.tellg(); //7, cursor chilo eof()  cout<<(char)fin.get(); //o  }  0H1I  0HI  4e  7o  61  7o |  |

Initializer

|  |  |
| --- | --- |
| #include <iostream>  using namespace std;  class Dummy{  private:  int a;  public:  Dummy()  {  a=5;  }  };  int main()  {  Dummy d1;  return 0;  } | #include <iostream>  using namespace std;  class Dummy{  private:  int a;  public:  Dummy():a(5)  {  }  void showdata()  {  cout<<a;  }  };  int main()  {  Dummy d1;  d1.showdata();  return 0;  } |
| #include <iostream>  using namespace std;  class Dummy{  private:  int a;  const int k; //ekane k er man assign kora jabe na. C++ er rule  public:  Dummy():k(5)  {  a=55; //egulote kono rule nai  }  };  int main()  {  Dummy d1;  return 0;  }  #include <iostream>  using namespace std;  class Dummy{  private:  int a;  const int k;  int &x;  public:  Dummy(int &n):a(5),k(6),x(n)  {  }  void showdata()  {  cout<<a;  cout<<k;  cout<<x;  }  };  int main()  {  int m=0;  Dummy d1(m); // m er reference disi  d1.showdata();  return 0;  }  560 | #include <iostream>  using namespace std;  class Dummy{  private:  int a;  const int k;  public:  Dummy():a(5),k(6)  {  }  void showdata()  {  cout<<a;  cout<<k;  }  };  int main()  {  Dummy d1;  d1.showdata();  return 0;  }  56 |

Deep copy & Shallow copy

Copy

|  |  |
| --- | --- |
| #include<iostream>  using namespace std;  class Dummy{  private:  int a,b;  public:  void setdata(int x, int y)  {  a=x,b=y;  }  void showdata()  {  cout<<a << "and" <<b;  }  };  int main()  {  Dummy d1;  d1.setdata(3,4);  Dummy d2=d1; //shallow copy sadiron vabe copy hoise  d1.showdata();  return 0;  } |  |

Type conversion

|  |  |  |  |
| --- | --- | --- | --- |
| Primitive to class  #include<iostream>  using namespace std;  class Complex{  private:  int a,b;  public:  Complex() //tai ami without argument ekta constructor banalam  {  }  Complex (int k) //Constructor banaise  {  a=k;b=0;  }  void setdata(int x, int y)  {  a=x; b=y;  }  void showdata()  {  cout<<a<<b;  }  };  int main()  {  Complex c1; // ekane Jodi empty constructor (jeta upore red color) na banai taile error dibe.  // karon compiler nije ekane constructor banabe na, jeta se always kore, ken korbe na  // karon ami nije ekane constructor create korsi  c1.setdata(3,4);  int z=5;  c1=z; //c1.Complex(z),  dan side e jeta liki seta constructor er argument  c1.showdata();  return 0;  } | | Class to primtive  #include<iostream>  using namespace std;  class Complex{  private:  int a,b;  public:  void setdata(int x, int y)  {  a=x; b=y;  }  void showdata()  {  cout<<a<<b;  }  operator int()  {  return(a);  } //casting operator  };  int main()  {  Complex c1;  c1.setdata(3,4);  int z;  z=c1;  c1.showdata();  cout<<z;  return 0;  } | |
| Class to class  #include<iostream>  using namespace std;  class product{ //age jodi item class define kortam tahole okane onek kisu requried jeta product class a ase  private:  int m,n;  public:  void setdata(int x, int y)  {  m=x;n=y;  }  int getM() //karon eta item class a pass korte hobe, kintu eta private tai return korlam  {  return m;  }  int getN()  {  return n;  }  };  class item{  private:  int a,b;  public:  item(){}  item(product p) //constructor ei class a banalam karon eta assignment er left side a ase ar constructor er maddome jodi ek class er data onno class  // a nite chai taile left side er class constructor banate hobe  {  a=p.getM(); //a=p.m; likle error karon m is pvt in another class  b=p.getN();  }  void show\_data()  {  cout<<a<<b;  }  };  int main()  {  item i1;  product p1;  p1.setdata(3,4);  i1=p1;  i1.show\_data();  return 0;  } | | |  |
| Exception handling  #include<iostream>  using namespace std;  int main()  {  cout<<"Welcome";  try{  throw 10;  cout<<"ae line cholbe na";  }  catch(int a){  cout<<"\nI am in catch and a is:"<<a;  }  cout<<"\nlast line";  }  Welcome  I am in catch and a is:10  last line  #include<iostream>  using namespace std;  int main()  {  cout<<"Welcome";  //try{  throw 10;  cout<<"ae line cholbe na";  //}  /\*  catch(int a){  cout<<"\nI am in catch and a is:"<<10;  }  \*/  cout<<"\nlast line";  }  Welcome terminate called after throwing an instance of 'int'  #include<iostream>  using namespace std;  int main()  {  cout<<"Welcome";  try{  throw 10;  cout<<"ae line cholbe na";  }  catch(double a){  cout<<"\nI am in catch and a is:"<<10;  }  catch(int a){  cout<<"\nI am in catch and a is:"<<10;  }  cout<<"\nlast line";  }  Welcome  I am in catch and a is:10  last line | #include<iostream>  using namespace std;  int main()  {  cout<<"Welcome";  try{  throw 10;  cout<<"ae line cholbe na";  }  /\* catch(int a){  cout<<"\nI am in catch and a is:"<<10;  }  \*/  cout<<"\nlast line";  }  ERROR try likle catch liktei hobe  #include<iostream>  using namespace std;  int main()  {  cout<<"Welcome";  /\*try{  throw 10;  cout<<"ae line cholbe na";  }  \*/  catch(int a){  cout<<"\nI am in catch and a is:"<<10;  }  cout<<"\nlast line";  }  ERROR catch likle try liktei hobe  Tai try and catch is a pair  #include<iostream>  using namespace std;  int main()  {  cout<<"Welcome";  try{  throw 10;  cout<<"ae line cholbe na";  }  catch(double a){  cout<<"\nI am in catch and a is:"<<a;  }  cout<<"\nlast line";  }  Welcometerminate called after throwing an instance of 'int' | | |

|  |  |
| --- | --- |
| #include<iostream>  using namespace std;  int main()  {  cout<<"Welcome";  int i=3;  try{  if(i==1)  throw 10;  if(i==2)  throw 20;  if(i==3)  throw 30;  }  catch(int a){  cout<<"\nI am in catch and a is:"<<a;  }  cout<<"\nlast line";  }  Welcome  I am in catch and a is:30  last line | #include<iostream>  using namespace std;  int main()  {  cout<<"Welcome";  int i=3;  try{  if(i==1)  throw 10;  if(i==2)  throw “h”i;  if(i==3)  throw 3.70;  }  catch(...){  cout<<"\nI am in catch and i is:"<<i;  }  cout<<"\nlast line";  }  Welcome  I am in catch and i is:3  last line |

Dynamic Constructor

|  |
| --- |
| #include <iostream>  using namespace std;  class A  {  int a;  public:  void f1()  {  cout<<"Hi";  }  virtual ~A(){}  };  class B: public A  {  int b;  public:  void f2()  {  cout<<"Hello";  }  ~B(){}  };  int fun()  {  A \*p = new B;  p->f1();  delete p;  }  int main()  {  fun();  return 0;  } |

ARRAY in STL

#include <iostream>

#include<array>

using namespace std;

int main()

{

array <int,8> data\_array={11,22,33,44,55};

cout<<"\n"<<data\_array.at(2);

//cout<<data\_array.at(7); //terminate:out of range

cout<<"\n"<<data\_array[2];

cout<<"\n"<<data\_array.front();

cout<<"\n"<<data\_array.back();

data\_array.fill(10);

for(int i=0;i<8;i++)

cout<<data\_array[i];

cout<<endl;

array <int,5> data\_array1={11,22,33,44,55};

array <int,5> data\_array2={66,77,88,99,98};

data\_array1.swap(data\_array2);

for(int i=0;i<5;i++)

cout<<data\_array1[i];

cout<<endl;

for(int i=0;i<5;i++)

cout<<data\_array2[i];

cout<<endl;

cout<<"Size of data\_array"<<data\_array.size();

return 0;

}

33

33

11

01010101010101010

6677889998

1122334455

Size of data\_array8

Pair

#include <iostream>

using namespace std;

int main()

{

pair <string,int> p1; //can be <string,string> or <int,float>etc

p1=make\_pair("Golam",32);

cout<<p1.first<<endl;

cout<<p1.second<<endl;

// <int,class>

class student{

private:

string name;

int age;

public:

void set\_data(string n, int a)

{

name=n;

age=a;

}

void show\_data()

{

cout<<name;

cout<<age;

}

};

student s2;

s2.set\_data("Dastoger",32);

pair <int,student> p2;

p2=make\_pair(404,s2);

cout<<p2.first;

student s3=p2.second;

s3.show\_data();

pair <string,int>p4;

p4=make\_pair("Golam",32);

if(p4==p1)

printf("they are same in value");

return 0;

}

Golam

32

404Dastoger32

TUPPLE

#include <iostream>

#include<tuple>

using namespace std;

int main()

{

tuple <string,int,int>t1;

t1=make\_tuple("Dhaka",1972,2019);

cout<<std::get<0>(t1)<<" ";

cout<<get<1>(t1)<<" "; //ekane std:: dei nai karon using namespace liksi

cout<<get<2>(t1)<<" ";

tuple<string,int,int>t2;

t2=make\_tuple("Bangladesh",2019,1972);

if(get<1>(t1) > get<1>(t2))

printf("error");

else

printf("\n1972 IS NOT GRATER THAN 2019");

return 0;

}

Dhaka 1972 2019

1972 IS NOT GRATER THAN 2019

Vector

#include <iostream>

#include<vector>

using namespace std;

int main()

{

vector <int>v1; //ekane capacity bydefault 1

cout<<v1.capacity()<<endl;;

vector<int>v2(5); //capacity size 5

vector<int>v3(5,10); // 5 ta array te e 10 raklam 10 | 10 | 10 | 10 | 10

cout<<v3[0]<<" "<<v3[1]<<" "<<v3[2]<<" "<<v3[3]<<" "<<v3[4]<<endl;

vector<string>v4(3);// 3 ta string nilam kisu rakar jonno

vector<string>v5(4,"hello");

cout<<v5[0]<<" "<<v5[1]<<" "<<v5[2]<<" "<<v5[3]<<endl;

vector <int> v6{10,20,30,40,50};

for(int i=0;i<5;i++)

cout<<v6[i];

//push\_back() er sahajje amra vector er sesh a j kono value add korte pari

v1.push\_back(10);

cout<<"\n" "Update capacity of v1 is "<<v1.capacity()<<endl;

for(int i=0;i<1;i++)

cout<<v1[i];

v1.push\_back(20);

cout<<"\n" "Update capacity of v1 is "<<v1.capacity()<<endl;

for(int i=0;i<2;i++)

cout<<v1[i];

v1.push\_back(20);

cout<<"\n" "Update capacity of v1 is "<<v1.capacity()<<endl;

for(int i=0;i<3;i++)

cout<<v1[i];

//pop\_back removes last element also reduce the array size

for(int j=0;j<10;j++)

v2.push\_back(10\*(j+1));

cout<<"\n" "Update capacity of v2 is "<<v2.capacity()<<endl;

return 0;

}

0

10 10 10 10 10

hello hello hello hello

1020304050

Update capacity of v1 is 1

10

Update capacity of v1 is 2

1020

Update capacity of v1 is 4

102020

Update capacity of v2 is 20