**1.Class and objects**

**#include<iostream>**

**using namespace std;**

**class book**

**{**

**private:**

**int bcode;**

**char title[50];**

**public:**

**void get()**

**{**

**cout<<"enter the book code,title of the book";**

**cin>>bcode;**

**cin>>title;**

**}**

**void put()**

**{**

**cout<<"book code"<<bcode;**

**cout<<"book title"<<title;**

**}**

**};**

**int main()**

**{**

**book b;**

**b.get();**

**b.put();**

**return 0;**

**}**

**2. Function overload**

**#include <iostream>**

**using namespace std;**

**void add(int a,int b)**

**{**

**cout<<"sum"<<(a+b);**

**}**

**void add(double a,double b)**

**{**

**cout<<"add"<<(a+b);**

**}**

**int main()**

**{**

**add(01,12);**

**add(1.6,3.2);**

**return 0;**

**}**

**3.string and exception**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**string s = "clow";**

**try**

**{**

**if(s == "clown")**

**{**

**throw(s);**

**cout<<"after throw/n";**

**}**

**else**

**{**

**cout<<"throw..";**

**}**

**}**

**catch(string s)**

**{**

**cout<<"caught"<<s;**

**}**

**return 0;**

**}**

**4.Templates**

**#include<iostream>**

**using namespace std;**

**template <class T>**

**class Number**

**{**

**private:**

**T num;**

**public:**

**Number(T n):num(n){}**

**T getNum()**

**{**

**return num;**

**}**

**};**

**int main()**

**{**

**Number<int>numberInt(7);**

**Number<double>numberDouble(7.7);**

**cout<<"intNumber="<<numberInt.getNum()<<endl;**

**cout<<"doubleNumber="<<numberDouble.getNum()<<endl;**

**return 0;**

**}**

**5.Virtual function using polymorphism**

#include <iostream>

using namespace std;

class Car

{

public:

void s()

{

cout<<"In base"<<endl;

}

};

class Bug : public Car

{

public :

void s()

{

cout<<"In derived"<<endl;

}

} ;

int main()

{

Bug bug;

Car\* c = &bug;

c->s();

return 0;

}

**6.constructor and destruct**

**#include <iostream>**

**using namespace std;**

**class car**

**{**

**string model;**

**string brand;**

**public:**

**car(string m,string b)**

**{**

**model = m;**

**brand = b;**

**}**

**~car()**

**{**

**cout<<"destuct";**

**}**

**void getbrand()**

**{**

**cout<<"model"<<model<<endl;**

**cout<<"brand"<<brand<<endl;**

**}**

**};**

**int main()**

**{**

**car c[6]={car("chiron","bugatti"),car("veyron\_SS","bugatti"),car("zonda","pagani"),car("r34","nissan"),car("venno","lamborghini"),car("LFA","lexus")**

**};**

**c[0].getbrand();**

**c[1].getbrand();**

**c[2].getbrand();**

**c[3].getbrand();**

**c[4].getbrand();**

**c[5].getbrand();**

**}**

**7.Inhetianca**

**2. Multilevel Inheritance**

#include<iostream>

using namespace std;

class arith\_op

{

public:

int a = 5;

int b = 30;

};

class add : public arith\_op

{

public:

void addtwonum()

{

cout<<a + b;

}

};

class mul : public add

{

public:

void multwonum()

{

cout<<a\*b;

}

};

int main()

{

mul Numbers;

cout<<"firstNumber:"<<Numbers.a<<endl;

cout<<"secondNumber:"<<Numbers.b<<endl;

cout<<"Adding two numbers:";Numbers.addtwonum();

cout<<"Multiply two numbers:";Numbers.multwonum();

}

**3.Hybrid Inheritance**

**#include <iostream>**

**using namespace std;**

**class student**

**{**

**public:**

**int id ;**

**char name;**

**void getstudent()**

**{**

**cout<<"Enter student Id,student name";**

**cin>>id>>name;**

**}**

**};**

**class marks : public student**

**{**

**protected:**

**int marks\_math,marks\_phy,marks\_chem;**

**public :**

**void getmarks()**

**{**

**cout<<"Enter 3 subject mark";**

**cin>>marks\_math>>marks\_phy>>marks\_chem;**

**}**

**};**

**class sports**

**{**

**public:**

**int spmarks;**

**void getsports()**

**{**

**cout<<"Enter sports marks";**

**cin>>spmarks;**

**}**

**};**

**class result : public marks , public sports**

**{**

**public:**

**int total\_marks;**

**float avg\_marks;**

**void display()**

**{**

**total\_marks = marks\_math+marks\_phy+marks\_chem;**

**avg\_marks = total\_marks/3.0;**

**cout<<"total marks ="<<total\_marks<<endl;**

**cout<<"Average marks ="<<avg\_marks<<endl;**

**cout<<"average + sports marks="<<avg\_marks+spmarks;**

**}**

**};**

**int main()**

**{**

**result res;**

**res.getstudent();**

**res.getmarks();**

**res.getsports();**

**res.display();**

**return 0;**

**}**