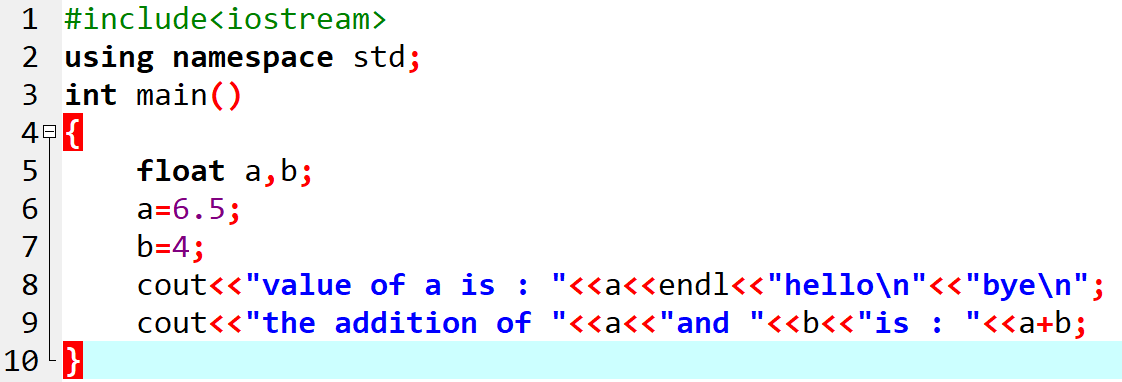
**1)**

#include<iostream>

using namespace std;

int main()

{

float a,b;

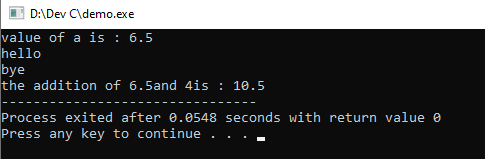
a=6.5;

b=4;

cout<<"value of a is : "<<a<<endl<<"hello\n"<<"bye\n";

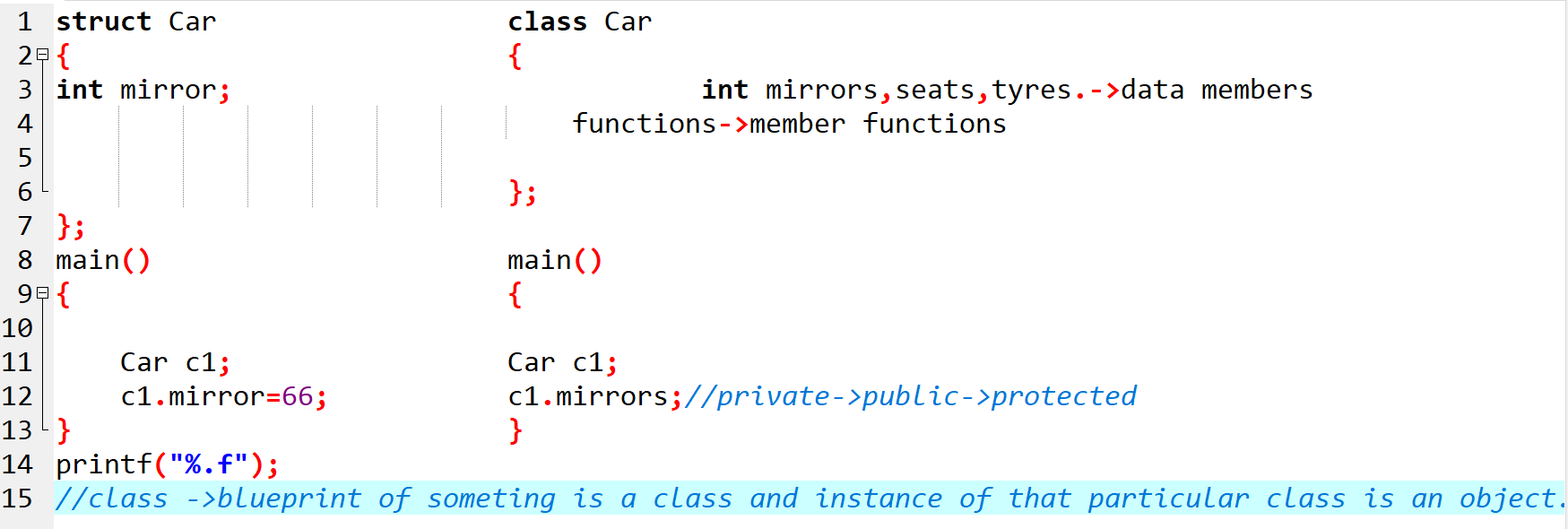
cout<<"the addition of "<<a<<"and "<<b<<"is : "<<a+b;

}

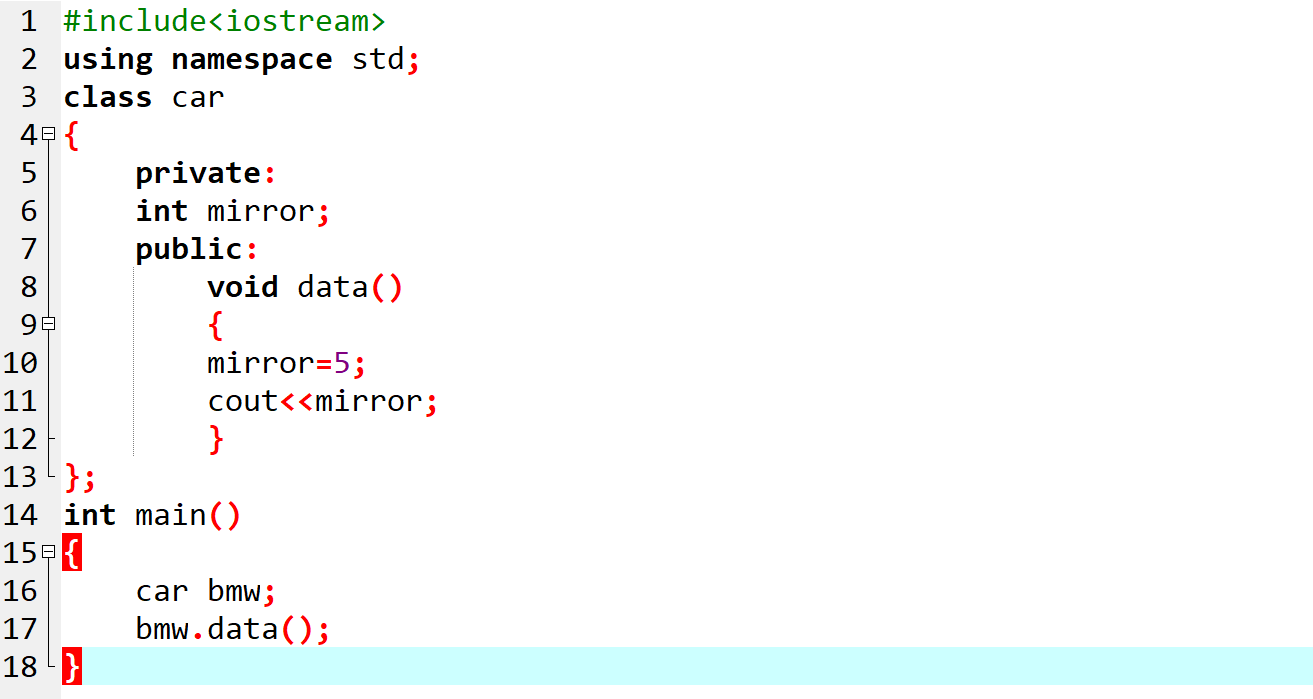


--------------------------------------------------------------------------------------------------------------------------------------------------------

**2)**



**3)**



#include<iostream>

using namespace std;

class car

{

private:

int mirror;

public:

void data()

{

mirror=5;

cout<<mirror;

}

};

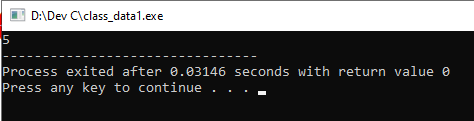
int main()

{

car bmw;

bmw.data();

}



----------------------------------------------------------------------------------------------------------------------------------------------

**4)**



#include<stdio.h>

#include<string.h>

int main()

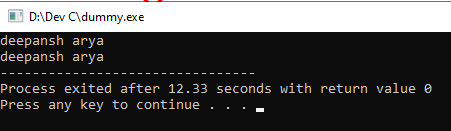
{

char a[10];

gets(a); //gets mean ‘get string’, it will not break string input on seeing space ,so it can store

printf("%s",a); //complete sentence in string array like name with sirname.

}



-------------------------------------------------------------------------------------------------------------------------------------------

**5)**

//Write a program to enter Employee first name, last name, Account number and various databases using single inheritance.

#include<iostream>

#include<bits/stdc++.h>

#include<string.h>

using namespace std;

class Office

{

protected:

char first\_name[10];

char last\_name[10];

long int account\_no;

};

class Employee:public Office

{

public:

void get\_Employee\_Info()

{

cout<<"Enter first name : ";

gets(first\_name);

cout<<"Enter last name : ";

cin>>last\_name;

cout<<"Enter account number : ";

cin>>account\_no;

}

void print\_Employee\_Info()

{

strcat(first\_name, last\_name);

cout<<"name is : "<<first\_name<<endl<<" and account number is : "<<account\_no<<endl;

}

};

int main()

{

Employee E1;

E1.get\_Employee\_Info();

E1.print\_Employee\_Info();

}

//int 2->turbo 2 bytes->4 bytes(dev)->-32768 to +32767

//float 4 bytes->32 bits->2^32

//int ->2 bytes=>16 bits=>2^16=>65535 unsigned ->%u

//int->long int

//float->long float(double(%lf))

//sizeof(int)