**\_\_\_\_\_**

1. Write the first line of the declaration for a class CAT that is publicly derived from a Class ANIMAL.

……………………………………………………………………………………………..

1. Assume a class **Derv** that is **privately** derived from class **Base**.

 An object of class Derv located in main() can access

1. public members of Derv.
2. protected members of Derv.
3. private members of Derv.
4. public members of Base.
5. protected members of Base.
6. private members of Base.
7. **Trace** the program then Define the differences between the function **overloading** and **overwritten**?

#include <iostream>

using namespace std;

////////////////////////////////////////////////////////////////

class base

{

private:

int privdata;

protected:

int protdata;

public:

int pubdata;

void function(){ privdata=protdata =pubdata= 0;

cout<< "base fun"<<endl;}

void function(int x, int y, int z)

{ privdata= x; protdata = y; pubdata= z; }

void print(){ cout<< "base Class"; }

void print(){ cout<< "base Class info are " <<privdata<<protdata <<pubdata<<endl;}

};//end class

////////////////////////////////////////////////////////////////

class derive : public base //publicly-derived class

{ int privatedataB;

public:

void function()

{ privatedataB= 0;cout<< "der fun"; }

void function(int x, int y, int z, int i)

{function();

privatedataB= i;}

//to print all class info

void print(){ cout<<"the class info are " <<privatedataB; }

};

////////////////////////////////////////////////////////////////

void main()

{base ob ;

ob.function();

derive obj;

obj.function(2,4,5,5);

obj.function();

obj.function(2,4,5);

}

1. Trace the following program and Find Errors and correct them:

#include <iostream>

using namespace std;

class Base

{ int m;

public:

 Base(int mValue){ m = mValue; cout<<"Constructing base"; }

};

class Derived: public Base

{double n;

public:

 Derived(double nValue) { n =nValue; cout<<"Constructing derived"; }

};

int main(){

 Base b1;

 Base b2(3.0);

 Derived d1

 Derived d2(4.0);

return 0;

}