**Question 1 ( give out put )**

struct CA

{

void fun() { cout<<"Apple"<<endl; }

};

struct Smart

{

CA \*pt;

Smart():pt(new CA()){}

CA \* operator->(){ return pt; }

void fun() {cout<<"Orange "<<endl; }

};

void main()

{

Smart \*p=new Smart();

p->fun();

}

**Question 2 ( give out put )**

struct CA

{

CA()

{

cout<<"Pine"<<endl;

this->fun();

}

virtual void fun() { cout<<"Apple"<<endl; }

};

struct CB:CA

{

CB()

{

this->fun();

}

void fun() { cout<<"Orange"<<endl; }

};

void main()

{

CB obj;

}

**Question 3 ( give out put )**

struct CA

{

static int count;

int i;

CA()

{

i=++count;

}

~CA()

{

cout<<i<<" "<<count<<endl;

}

static void sFun()

{

count++;

}

};

int CA::count=1;

void main()

{

CA::sFun();

CA obj1;

CA obj2;

CA obj3;

}

**Question 4 ( give out put )**

struct CA

{

static int count;

int i;

CA()

{

i=++count;

}

CA(const CA&ob):i(ob.i)

{

cout<<"Copy"<<endl;

}

~CA()

{

cout<<i<<" "<<count<<endl;

count++;

}

static void sFun()

{

count++;

}

};

int CA::count=1;

void main()

{

try

{

throw \*new CA();

}

catch(CA Ob)

{

}

}

**Question 5 ( give out put )**

struct CA

{

static int count;

int i;

CA()

{

i=++count;

}

CA(const CA&ob):i(ob.i)

{

cout<<"Copy"<<endl;

}

~CA()

{

cout<<i<<" "<<count<<endl;

count++;

}

static void sFun()

{

count++;

}

};

int CA::count=1;

CA\* fun()

{

return new CA();

}

void main()

{

try

{

throw fun();

}

catch(CA Ob)

{

cout<<"apple"<<endl;

}

Anwer Question 1:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Anwer Question 2:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Anwer Question 3:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Anwer Question 4:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Anwer Question 5:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_