/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online C++ Compiler.

Code, Compile, Run and Debug C++ program online.

Write your code in this editor and press "Run" button to compile and execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Online C++ Compiler.

Code, Compile, Run and Debug C++ program online.

Write your code in this editor and press "Run" button to compile and execute it.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

using namespace std;

class DSArray

{

private:

int max,curr;

int arr[10];

public:

DSArray()

{

curr=0;

max=10;

}

void addelement(int value)

{

if(curr==max)

{

cout<<"limit reached";

return;

}

else{

arr[curr]=value;

curr++;

cout<<"element added";

}

print();

}

void print()

{

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

{

cout<<arr[i]<<endl;

}

}

void insert\_beg(int value)

{curr++;

for(int i=curr;i>=0;i--)

{

arr[i]=arr[i-1];

}

arr[0]=value;

}

void insert\_end(int value)

{arr[curr]=value;

curr++;

}

void insert\_pos(int value,int pos)

{

for(int i=curr;i>pos-1;i--)

{

arr[i]=arr[i-1];

}

arr[pos-1]=value;

curr++;

}

void del\_last()

{if(curr==0)

cout<<"list is empty";

else

curr--;

// arr[curr-1]='\0';

}

void del\_first()

{if(curr==0)

cout<<"list is empty";

else

{int i;

for( i=1;i<curr;i++)

{

arr[i-1]=arr[i];

}

curr--;

arr[i]='\0';

}

}

void del\_pos(int pos)

{if(curr==0)

cout<<"list is empty";

else

{int i;

for( i=pos-1;i<curr-1;i++)

{

arr[i]=arr[i+1];

//curr--;

}

curr--;

}

}

};

int main()

{

int i,choice;

i=0;

DSArray obj;

while(i!=1)

{

cout<<"linear array menu"<<endl;

cout<<"press 1 to add element"<<endl;

cout<<"press 2 to show element"<<endl;

cout<<"press 3 to exit"<<endl;

cout<<"press 4 to delete from last"<<endl;

cout<<"press 5 to delete from first"<<endl;

cout<<"press 6 to delete from pos"<<endl;

cout<<"press 7 to insert in begining"<<endl;

cout<<"press 8 to insert in end"<<endl;

cout<<"press 9 to insert in any pos"<<endl;

cout<<"enter your choice"<<endl;

cin>>choice;

switch(choice)

{

case 1:

int value;

cout<<"enter value";

cin>>value;

obj.addelement(value);

break;

case 2:

cout<<endl<<"elements are";

obj.print();

break;

case 3:

i=1;

break;

case 4:

obj.del\_last();

break;

case 5:

obj.del\_first();

break;

case 6:

int pos;

cout<<"enter pos";

cin>>pos;

obj.del\_pos(pos);

break;

case 7:

int value1;

cout<<"enter value"<<endl;

cin>>value1;

obj.insert\_beg(value1);

break;

case 8:

int value2;

cout<<"enter value to enterin an end"<<endl;

cin>>value2;

obj.insert\_end(value2);

break;

case 9:

int value3,pos3;

cout<<"enter value and pos"<<endl;

cin>>value3>>pos3;

obj.insert\_pos(value3,pos);

break;

default:

cout<<"invalid choice";

}

}

return 0;

}

==================================================================================If your question intended to speak about main() method from where JVM execution begins, then yes are right main() method isn’t inherited or overridden, because, main() method has static keyword which tells that it belongs to class rather to any specific instance

In simple words, a constructor cannot be inherited, since in subclasses it has a different name (the name of the subclass).

class A {

A();

}

class B extends A{

B();

}

You can do only:

B b = new B(); // and not new A()

Methods, instead, are inherited with "the same name" and can be used.

As for the reason: It would not have much sense to inherit a constructor, since constructor of class A means creating an object of type A, and constructor of class B means creating an object of class B.

You can still **use** constructors from A inside B's implementation though:

class B extends A{

B() { super(); }

}