Topics: Memory Allocation, Relationships in OOPS, Class and members

Q1. Analyze the correctness and output of following programs

#include <iostream>
#include <malloc.h>
using namespace std;
int main() {
float *a;
a = (float *)malloc(sizeof(int));
a[0] = 4.5;
cout<<a[0];
return 0;
}
	#include <iostream>
	#include <malloc.h>
	using namespace std;
	int main() {
	int *a;
	a = (int *)malloc(sizeof(float));
	a[0] = 5;
	cout<<a[0];
	return 0;
	}

#include <iostream>
#include <malloc.h>
using namespace std;
int main() {
int *a, *b;
a = (int *)malloc(sizeof(int));
b = (int *)malloc(5*sizeof(int));
cout<<sizeof(a)<< sizeof(b);
return 0;
}
	#include <iostream>
	#include <malloc.h>
	using namespace std;
	int main() {
	int *a;
	a[0] = (int *)malloc(sizeof(int));
	a[0] = 5;
	cout<<a[0];
	return 0;
	}

#include <iostream>
#include <malloc.h>
using namespace std;
int main() {
int *a[5];
a[0] = (int *)malloc(sizeof(int));
a[0][0] = 5;
cout<<a[0][0];
return 0;
}
	#include <iostream>
	#include <malloc.h>
	using namespace std;
	int main() {
	struct node{int a[10];};
	struct node *n;
	n = (struct node *)malloc(sizeof(struct node));
	cout<<sizeof(n);
	return 0;
	}

#include <iostream>
#include <malloc.h>
using namespace std;
int main() {
int *a[5];
a[0] = (int *)malloc(2*sizeof(int));
a[0][1] = 5;
cout<<a[0][1];
return 0;
}
	#include <iostream>
	#include <malloc.h>
	using namespace std;
	int main() {
	int *a = (int *)malloc(5*sizeof(int));
	a[0] = 1; a[1] = 2; a[2] = 3; a[3] = 4; a[4] = 5;
	delete(a);
	cout<<a[0]<<a[1]<<a[2]<<a[3]<<a[4];
	return 0;
	}

Q2. You are required to create a class — "Invoice". This class might be used by a departmental store to represent an invoice for an item sold at the store. An Invoice should include four data members—item number (type string), item description or name (type string), quantity of the item being purchased (type int) and price per item (type int). Your class should have a constructor that initializes the four data members. Provide a set and a get function for each data member. In addition, provide a member function named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as an int value. The class should be able to give the count of all the objects which are created through copy operation. Write a function that accepts two invoices for different departmental stores and return the maximum quantity out of two invoices. Write a test program that demonstrates class Invoice's capabilities.

Q3. Create a class —Employee that includes three pieces of information as data members—a first name (type string), a last name (type string) and a monthly salary (type int). Your class should have a constructor that initializes the three data members. Provide a set and a get function for each data member. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10 percent raise and display each Employee's yearly salary again. The class should be able to give the count of all the default objects. Write a function which takes two employees and return the name of the employee with higher salary. Write a test program that demonstrates class Employee capabilities.

Q4. Identify the relationship(s); analyze the sequence of constructor and destructor invocation; and obtain the output

class ABC	class KLM	class XYZ

{
int x;
public:
ABC() { cout<<"1"; }
~ABC() { cout<<"2"; }
};
	{
	int y;
	ABC O1;
	public:
	KLM() { cout<<"3"; }
	~KLM() { cout<<"4"; }
	};
		{
		int y; XYZ O2;
		ABC O3;
		public:
		XYZ() { cout<<"5"; }
		~XYZ() { cout<<"6"; }
		};

int main() { XYZ O4, *O5; KLM *O6; O5 = new XYZ; O6 = new XYZ; delete(O5); return 0; }		