14Spring CS 31 Sec1B TA: Zhou Ren

Examples of Pointers and Structure

Example #1: What is the output of this program below?

```
#include <iostream>
using namespace std;
int main()
{
	int value = 1000;
	int *p = &value;
	int *q;
	q = &value;

	cout << "Address of value is: " << &value << endl;
	cout << "Value of value is: " << value << endl;

	cout << "value of p is: " << p << endl;
	cout << "address of p is: " << &p << endl;
	cout << "Value of variable p references to is: " << *p << endl;

	cout << "value of q is: " << q << endl;
	cout << "address of q is: " << &q << endl;
	cout << "Value of variable q references to is: " << *q << endl;

	return 0;
}
```

Example #2: What is the output of this program below?

```
#include <iostream>
using namespace std;

int main()
{
	int val;
	int *p = &val;
	*p = 100;

	cout << p << endl;
	cout << *p << endl;

	return 0;
}
```

Example #3: What is the output of this program below?

```
#include <iostream>
using namespace std;
int main()
{
	int a = 5;
```

```
        int *p = &a;

        //cin >> *p;
        //cin >> *(&a);

        cout << p << endl;
        cout << *p << endl;
        cout << &(*p) << endl;
        cout << *(&(*p)) << endl;
        cout << &(*(&(*p))) << endl;
        cout << *(&(*(&(*p)))) << endl;

        return 0;
}
```

Example #4: What is the output of this program below?

```
#include <iostream>
using namespace std;

int main()
{


        int i;
        int *p = &i;
        *p = 100;

        cout << p << endl;
        cout << *p << endl;

        *p = *p + 7;

        cout << p << endl;  //
        cout << *p << endl;  //

        return 0;
}
```

Example #5: What is the output of this program below?

```
#include <iostream>
using namespace std;

int main()
{
        int v1 = 99, v2=1000;

        cout << "Address of v1 is: " << &v1 << endl;
        cout << "Address of v2 is: " << &v2 << endl;
        cout << "Value of v1 is: " << v1 << endl;
        cout << "Value of v2 is: " << v2 << endl;
        cout << "----------------------------- " << endl;
```

```
        int *p1 = &v1;
        int *p2 = &v2;

        cout << "value of p1 is: " << p1 << endl;
        cout << "value of p2 is: " << p2 << endl;
        cout << "Value of the variable p1 points to is: " << *p1 << endl;
        cout << "Value of the variable p2 points to is: " << *p2 << endl;
        cout << "------------------------------ " << endl;

    *p1 = *p2;

        cout << "After running *p1 = *p2 ..." << endl;

        cout << "Address of v1 is: " << &v1 << endl;
        cout << "Address of v2 is: " << &v2 << endl;
        cout << "Value of v1 is: " << v1 << endl;
        cout << "Value of v2 is: " << v2 << endl;
        cout << "------------------------------ " << endl;

        *p1 = 199;
        v2  = 30000;

        cout << "After running *p1 = 199 and v2 = 30000" << endl;

        cout << "Value of p1 is: " << p1 << endl;
        cout << "Value of p2 is: " << p2 << endl;
        cout << "Value of the variable p1 points to is: " << *p1 << endl;
        cout << "Value of the variable p2 points to is: " << *p2 << endl;
        cout << "------------------------------ " << endl;

        p1 = p2;

        cout << "After running p1 = p2 ..." << endl;

        cout << "Address of v1 is: " << &v1 << endl;
        cout << "Address of v2 is: " << &v2 << endl;
        cout << "Value of v1 is: " << v1 << endl;
        cout << "Value of v2 is: " << v2 << endl;

        cout << "Value of p1 is: " << p1 << endl;
        cout << "Value of p2 is: " << p2 << endl;
        cout << "Value of the variable p1 points to is: " << *p1 << endl;
        cout << "Value of the variable p2 points to is: " << *p2 << endl;
        cout << "------------------------------ " << endl;

        return 0;
}
```

Example #6: Does the following program have compilation error?

```
#include <iostream>
using namespace std;
int main()
{
        int a;
```

```
        int *p = &a;

        cin >> p;
        //cin >> *p;
        //cin >> *(&a);

        cout << *p << endl;

        return 0;
}
```

Example #7: What is the output of the program below?

```
#include <iostream>

using namespace std;

int main()
{
        int *a,*b,*c,*d,*e,*f,*g,*h,*i,*j,*k;

        *a=*b=*c=*d=*e=*f=*g=*h=*i=*j=*k=1;

        cout << *a+*b+*c+*d+*e+*f+*g+*h+*i+*j+*k << endl;

        return 0;
}
```

Example #8: Any Error here???

```
#include <iostream>

using namespace std;

int main()
{
        // The value of pi
        const double pi = 3.14159;
        const double *p = &pi;          // Hmmmm…?

        cout << *p << endl;

        return 0;
}
```

Example #9: Looks tricky but very important!!

```cpp
#include <iostream>
using namespace std;

struct Array
{
	int array[5];
};

double & GetWeeklyHours()
{
    double h = 46.50;
    double &hours = h;

    return hours;
}
//-------------------------------------------------------------------------
double * GetSalary()
{
    double salary = 26.48;
    double *HourlySalary = &salary;

    return HourlySalary;
}

Array* GetArray()
{
	Array a;
	a.array[0] = 0;
	a.array[1] = 1;
	a.array[2] = 2;
	a.array[3] = 3;
	a.array[4] = 4;

	return &a;
}
//-------------------------------------------------------------------------
int main()
{
    double hours  = GetWeeklyHours();
    double* psalary = GetSalary();
	double salary = *GetSalary();
	Array *pA = GetArray();
	Array A = *GetArray();

    cout << "Weekly Hours:  " << hours << endl;
    cout << "Hourly Salary: " << *psalary << endl;
	cout << "Hourly Salary: " << salary << endl;

	cout << "elements in pA\n";
	for(int i=0; i<5; i++)
		cout << pA->array[i] << "\t";
	cout << "elements in A\n";
	for(int i=0; i<5; i++)
		cout << A.array[i] << "\t";

    return 0;
}
```

Example #10: What is the output of the program below?

```
#include <iostream>
using namespace std;

int main()
{
	int x = 100;
	int *px = &x;

	cout << (*px)++ << endl;
	cout << *(px++) << endl;
	cout << *px << endl;
	cout << x << endl;

	return 0;
}
```

Example #11: Array is a consecutive storage space, so if we can use a pointer pointing to the beginning of the array, we can then use this pointer to traverse the whole array as long as the pointer points to the right place.
What is the output of the program below?

```
#include <iostream>
using namespace std;

int zeroNegative(double *x, int n)
{
	int count=0;
	for(int i = 0 ; i < n ; i++)
		if(*(x+i) < 0)  // equivalent to x[ i ]
		{
			*(x+i) = 0;     // equivalent to x[ i ]
			count++;
		}
	return count;
}

int main()
{
	double x[4] = { -1.1, 3, -300.80, 4 };

	cout << zeroNegative(x, 4) << endl;

	for(int i=0;i<4;i++)
		cout << x[ i ] << " ";
	cout << endl;
}
```

Example #12: Rewrite example #13

```
#include <iostream>
using namespace std;

int zeroNegative(double *x, int n)
{
	int count=0, i;
	double *p;
	for(i = 0, p = x ; i < n ; i++, p++)
		if(*p < 0)
		{
			*p = 0;
			count++;
		}
	return count;
}

int main()
{
	double x[4] = { -1.1, 3, -300.80, 4 };

	cout << zeroNegative(x, 4) << endl;

	for(int i=0;i<4;i++)
		cout << x[ i ] << " ";
	cout << endl;
}
```

Example #13: Rewrite the function strlen with pointer?

```
#include <iostream>
using namespace std;
int mystrlen(char *p)
{
	int len=0;
	while(*p++ != '\0') len++;
	return len;
}
int main()
{
	char str1[] = "CS31";
	char str2[] = "Pointers are very powerful!";
	cout << mystrlen(str1) << endl;
	cout << mystrlen(str2) << endl;
}
```

Example #14: Please rewrite strcpy function by using pointers.

```
 // strcpy
#include <iostream>
using namespace std;
```

```
void mystrcpy(char *str1, char *str2)
{
	//
}
int main()
{
	char str1[20] = "CS31Session1C";
	char str2[20] = "NoClassNextFriday";

	cout << "before copying" << endl;
	cout << str1 << endl << str2 << endl;

	cout << endl << "After copying" << endl;
	mystrcpy(str1,str2);
	cout << str1 << endl << str2 << endl;
	return 0;
}
```

Example #15: Please rewrite strcmp function by using pointers.

```
#include <iostream>
using namespace std;

// return 0 if equal
// return negative value if str1 is less than str2
// return positive value if str1 is greater than str2
int mystrcmp(char *str1, char *str2)
{

}
int main()
{
	char str1[20] = "CS31";
	char str2[20] = "A";

	cout << str1 << endl << str2 << endl;
	cout << mystrcmp(str1,str1) << endl;
	cout << mystrcmp(str1,str2) << endl;
	cout << mystrcmp(str2,str1) << endl;
	return 0;
}
```

Example #16: Example of struct

```
#include <iostream>
#include <string>
using namespace std;

struct employee
{
	int ID;
	double salary;
	string jobtitle;
};   // <- Please remember to put semicolon here….
```

```
int main()
{
	employee emp;

	cout << "Please enter employee information:\n";

	cout << "ID Number: ";
	cin >> emp.ID;

	cout << "Salary: ";
	cin >> emp.salary; cin.ignore(1000,'\n');

	cout << "Job title: ";
	getline(cin,emp.jobtitle);

	cout << "There are 1 employee:\n";
	cout << "ID Number: " << emp.ID << endl;
	cout << "Salary: " << emp.salary << endl;
	cout << "Job Title: " << emp.jobtitle << endl;

	return 0;
}
```

Example #17: Struct with pass by value/reference

```
#include <iostream>
#include <string>
using namespace std;

struct employee
{
	int ID;
	double salary;
	string jobtitle;
};

void getInput(employee emp)
{
	cout << "Please enter employee information:\n";
	cout << "ID Number: "; cin >> emp.ID;
	cout << "Salary: ";    cin >> emp.salary; cin.ignore(1000,'\n');
	cout << "Job title: "; getline(cin,emp.jobtitle);
}

int main()
{
	employee emp = {0,0,""};  // initialization
	// same as: emp.ID = 0; emp.salary = 0; emp.jobtitle = "";

	getInput(emp);

	cout << "There are 1 employee:\n";

	cout << "ID Number: " << emp.ID << endl;
	cout << "Salary: " << emp.salary << endl;
```

```
	cout << "Job Title: " << emp.jobtitle << endl;

	return 0;
}
```

Example #18: Structure Array, how to use pointer to do this??

```
#include <iostream>
#include <string>
using namespace std;

struct employee
{
	int ID;
	double salary;
	string jobtitle;
};

int main()
{
	employee emp[2];
	cout << "Please enter employee information:\n";

	for(int i=0;i<2;i++)
	{	cout << "ID Number: "; cin >> emp[i].ID;
		cout << "Salary: ";    cin >> emp[i].salary; cin.ignore(1000,'\n');
		cout << "Job title: "; getline(cin,emp[i].jobtitle);
	}

	cout << "There are 2 employees:\n";

	for(int i=0;i<2;i++)
	{
		cout << "Employee #" << i+1 << ":" << endl;
		cout << "ID Number: " << emp[i].ID << endl;
		cout << "Salary: " << emp[i].salary << endl;
		cout << "Job Title: " << emp[i].jobtitle << endl;
	}


	cout << "here, I am using pointer to a struct" << endl;
	employee* pe;
	for(int i=0;i<2;i++)
	{
		pe = emp + i;
		cout << "Employee #" << i+1 << ":" << endl;
		cout << "ID Number: " << pe->ID << endl;
		cout << "Salary: " << pe->salary << endl;
		cout << "Job Title: " << pe->jobtitle << endl;
	}

	return 0;
}
```

Example #19: Structure Array with Pass by Value/Reference

```
#include <iostream>
```

```
#include <string>
using namespace std;

struct employee
{
	int ID;
	double salary;
	string jobtitle;
};
void getInput(employee emp1[2]) // same as employee *emp
{
	cout << "Please enter employee information:\n";

	for(int i=0;i<2;i++)
	{	cout << "ID Number: "; cin >> emp1[i].ID;
		cout << "Salary: ";   cin >> emp1[i].salary; cin.ignore(1000,'\n');
		cout << "Job title: "; getline(cin,emp1[i].jobtitle);
	}
}
int main()
{
	employee emp[2];

	getInput(emp);

	cout << "There are 2 employees:\n";

	for(int i=0;i<2;i++)
	{
		cout << "Employee #" << i+1 << ":" << endl;
		cout << "ID Number: " << emp[i].ID << endl;
		cout << "Salary: " << emp[i].salary << endl;
		cout << "Job Title: " << emp[i].jobtitle << endl;

	return 0;
}
```