
Lab. (2)

Pointers and References

Answer the following questions

Question One:

Write a program in C++ to print the address and value using & (address of) and *(value at address) operators for integer variable (x=10) and double variable (y=3.4).

Question Two:

Write a program to store elements in an array and print the elements using pointer where x = [1, 5, 3, 9, 11, 7].

```
int main() {  
    int *ip;  
    int arr[] = { 10, 34, 13, 76, 5, 46};  
    ip = arr;  
    for (int x = 0; x < 6; x++)  
        cout << *ip << endl;  
        ip++;  
    {  
        return 0;  
    }
```

Question Three:

Write a program to swap two elements using pointer and functions.

```
#include <iostream>  
using namespace std;  
void test(int*, int*);  
int a = 5, b = 7;  
int main() {  
    cout << "Before changing:" << endl;  
    cout << "a = " << a << endl;  
    cout << "b = " << b << endl;  
  
    test(&a, &b);  
  
    cout << "\nAfter changing" << endl;  
    cout << "a = " << a << endl;  
    cout << "b = " << b << endl;  
    return 0;  
}
```

```

}

void test(int* n1, int* n2) {
    int c=a;
    *n1 = b;
    *n2 = c;
}

```

Question Four:

Write a program to shows elements value in an array and print the address using pointer where x = [10,20,30,40].

```

#include <iostream>
using namespace std;
int main()
{
int x[]={10,20,30,40};
int *ptr=x;

cout << "value " << *(ptr) << " has address of "<<ptr <<'\n';
cout << "value " << *(ptr+1)<< " has address of "<<ptr+1<<'\n';
cout << "value " << *(ptr+2)<< " has address of "<<ptr+2<<'\n';
cout << "value " << *(ptr+3)<< " has address of "<<ptr+3<<'\n';
return 0;
}

```

```

value 10 has address of 0x6ffe20
value 20 has address of 0x6ffe24
value 30 has address of 0x6ffe28
value 40 has address of 0x6ffe2c

```

Question Five: Choose the correct answer

- The _____, also known as the address operator, returns the memory address of a variable.
 - Asterisk (*)
 - Ampersand (&)
 - Percent Sign (%)
 - Exclamation Point (!)
- Which reference modifier is used to define reference variable?
 - &
 - \$
 - #
 - None of the mentioned
- The operator used for pointer dereferencing or indirection is ____
 - *
 - &
 - >
 - >>
- Choose the right option about the meaning of string* x, y;
 - y is a pointer to a string, x is a string
 - x is a pointer to a string, y is a string
 - both x and y are pointer to string types
 - none of the mentioned

Question six: Trace and find the following programs output

1)

```
1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5.     int z= 19;
6.     int *pointer1 = &z;
7.     int *pointer2 = pointer1;
8.     cout<<pointer1<<" "<<pointer2;
9.     return 0;
10. }
```

Assume the address of value z is 004eff08

2)

```
1. #include <iostream>
2. #include <string>
3. #include <cmath>
4. using namespace std;
5. int main()
6. {
7.     int x = 100;
8.     int y = 200;
9.     int* ptrnum = &x;
10.    y = sqrt(x)* y;
11.    *ptrnum = pow(sqrt(x),3);
12.    x *= 4;
13.    cout<< y<<" "<<x<<endl;
14.    return 0; }
```

3)

```
1. #include <iostream>
2. using namespace std;
3. int main() {
4.     int a = 9;
5.     int & aref = a;
6.     a++;
7.     cout << "The value of a is " << aref;
8.     return 0; }
```

4)

```
1. #include <iostream>
2.     using namespace std;
3.     int main() {
4.         int arr[] = {4, 5, 6, 7};
5.         int *p = (arr + 1);
6.         cout << *p;
7.         return 0; }
```

5)

```
1. #include <iostream>
2. using namespace std;
3. int main (){
4. int numbers[5];
5. int * p;
6. p = numbers; *p = 10;
7. p++; *p = 20;
8. p = &numbers[2]; *p = 30;
9. p = numbers + 3; *p = 40;
10. p = numbers; *(p + 4) = 50;
11. for (int n = 0; n < 5; n++)
12. cout << numbers[n] << ",";
13. return 0; }
```

6)

```
1. #include <iostream>
2. using namespace std;
3. double function(double *a);
4. int main() {
5. double y[2][3]={{13, 2.5, 3.0}, {4.4, 5.6, 6.7}};
6. double *dptr=&y[0][0];
7. cout << "Result = " << function(dptr) << endl;
8. return 0;
9. }
10. double function(double *a)
11. {
12. double result = 0.0;
13. int k, k_count=6;
14. for (k=0; k<=k_count-1; k++)
15. {
16. result += *(a+k);
17. }
18. return result; }
```