Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Quiz and Practice Questions on Pointers

Recap Quiz

IIT Bombay

Q1. Consider the following function call

```
int a, b;
cin >> a >> b;
func(&a, &b);
```

Parameters are passed here by:

A. Reference B. Value

C. Neither reference nor value

Recap Quiz

Q2. If a function f1 calls a function f2, which of the following are possible:

A. f2 can change values of local variables of f1

B. f2 cannot change values of local variables of f1

C. f1 can change values of local variables of f2 other than the formal parameters of f2

D. None of the above

Recap Quiz

Q3. Which of the following are true of function calls in C++ ?

A. Can accept pointers as parameters

B. Can return a pointer

C. Cannot dereference pointers passed as parameters

D. Cannot return a pointer if pointers are passed as parameters

Recap Quiz

Q4. A "swap" function that swaps the values of two integer variables in the calling function can be implemented by passing parameters by

A. Reference

B. Value (could be value of a pointer)

C. Both A and B

D. None of the above

Recap Quiz

Q5. A function f1 returns a pointer to a local variable of f1. If f2 calls f1 and tries to dereference the pointer returned by f1, which of the following may happen?

A. Memory access error during runtime

B. No error during runtime

C. Reading of garbage value by f2

D. None of the above

Practice Question 1

IIT Bombay

- **Consider the following code snippet of a function in C++:**

```
int * func(int * a, int b)  {
   int c = 0;
   if ((*a) == b)  return a;
   else return func(&c, c);
}
```

Practice Question 1

- **We want to invoke func from main as follows:**

```
int *p;
int q = ???;
p = func(&q, 0);
cout << *p;
```

What value of ??? will cause dereferencing of a bad address when trying to print *p ?

What value of ??? will avoid the above problem? What is the value output in this case?

Practice Question 2

Recall how we used 1-D arrays to store data in such a way that we could access the data as a 2-D array.

For example, in order to access a 10 x 10 array of int, we used:

```
int a1[10], a2[10], … a10[10];
int * A[10] = {&(a1[0]), &(a2[0]), …  &(a10[0])};
```

Practice Question 2

IIT Bombay

We now wish to write a function

int ** get2DArray(int n1, int n2) that can be used as shown below.

```
int ** A;
A = get2DArray(10, 10);
for (int i = 0; i < 10; i++)
   for (int j = 0; j < 10; j++)
        (A[i])[j] = i + j;
```

Practice Question 2

- **You may assume that you have access to the following black-box (unknown implementation) functions:**

  int * get1DIntArray(int n)

  int ** get1DIntPtrArray(int n);

Each of these functions may be assumed to return a pointer to an array that can be accessed even after the function returns.