Object Oriented Programming (CS1004)

Sessional-I Exam

Total Time: 1 Hour
Total Marks: 40
Total Questions: 02

Date: Feb 27, 2024

Course Instructor(s)
Mr. Aamir Rahim
Ms. Anosha Khan
Ms. Arooj Khalil
Ms. Samin Iftikhar
Mr. Uzair Naqvi
Mr. Waqas Manzoor

17.5

Semester: SP-2024
Campus: Lahore
Dept: FAST School of Computing

Nabiha Noor Khalique 23L-0967 2D
Student Name Roll No Section Student Signature

Vetted by Vetter Signature

IMPORTANT INSTRUCTIONS: Answer in the space provided. **Answers written on rough sheet will not be marked.** Do not use pencil or red ink to answer the questions. In case of confusion or ambiguity make a reasonable assumption.

12.5

CLO # 4: Apply good programming practices

Q1: [4x5 = 20 marks] Short Questions

Part (a) Write output of the code segment below. (There is no syntax error in the code.)

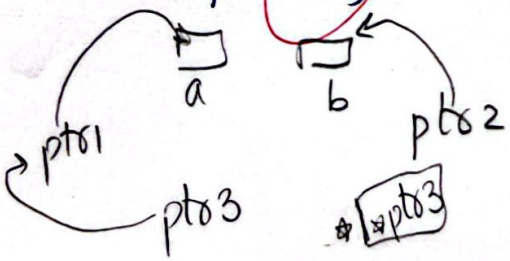
```
#include <iostream>
using namespace std;

void Swap(int*& a, int*& b)
{
        int* temp = a;
        a=b;
        b=temp;
}
```

```
void main()
{
        int a=5;
        int b=10;
        int* ptr1 = &a;
        int* ptr2 = &b;
        int** ptr3 = &ptr1;
        cout<<"Data = "<<**ptr3<<endl;
        int* temp1 = ptr1;
        int* temp2 = ptr2;
        Swap(temp1, temp2);
        cout<<"-------------"<<endl;
        cout<<"*ptr1 = "<<*ptr1<<endl;
        cout<<"*ptr2 = "<<*ptr2<<endl;
}
```

Output:

Data = 5 ;
*ptr1 = 10 ;
*ptr2 = 5 ;

2.5



Part (b): Write output of the code segment below. If there is any error, clearly mention the error. (There is no syntax error in this code.)

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

int* SomeFunction()
{
        int abc = 50;
        return &abc;
}

void main()
{
        int* ptr1 = SomeFunction();
        cout<<"Data = ";
        cout<<*ptr1<<endl;
}
```

ptr1

Output/Error:

Data = 50 ;

2

Part (c) Write the output of the code segment given below. (There is no syntax error in this code.)

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

void SomeFunction(int* arr, int size) {
    int* ptr1 = arr;
    int* ptr2 = arr + size - 1;
    while(ptr1 < ptr2) {
        *ptr1 = *ptr2;
        ptr1 = ptr1+2;
        ptr2--;
    }
}
```

```
int main() {
    int nums[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int* ptr = nums;
    SomeFunction(ptr, 10);
    for(int i = 0; i < 10; ++i) {
        cout << nums[i] << " ";
    }
    return 0;
}
```

ptr nums

Output:

10 2 9 4 8 6 7 8 9 10

Part (d) For the code segment given below, write output/error. In case of crash, highlight the line where program will crash. (There is no syntax error in this code.)

[THIS QUESTION IS NOT FOR BCS-2C]

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

int* GetData(int xyz)
{
        int* ptr = 0;
        if(xyz%2 == 0)
        {
                ptr = new int[5];
                for(int i=0; i<5; i++)
                        ptr[i] = i+1;
        }
        return ptr;

}
```

1 2345

```
int main() {
        int* array1[10];
        for(int i=0 ; i<10 ; i++)
        {
                array1[i] = GetData(i);
        }
        for(int i=0; i<10; i++)
        {
                for(int j=0; j<5 ; j++)
                {
                        array1[i][j] = array1[i][j] *2;
                        cout<<array1[i][j]<<" ";
                }
                cout<<endl;
        }
        //Assume we have Deallocation code here that
//successfully deallocates the memory.
}
```

0

It is one D-array.

Output/Error:

~~array1 is a 1D integer array.~~

0
0
2 4 6 8 10
0
2 4 6 8 10
0

2 4 6 8 10
0
2 4 6 8 10
0
2 4 6 8 10.

3

Part (d) ***[FOR BCS-2C ONLY]***

Consider the following program, **give C++ code for the class Point.** The distance formula is d = sqrt(dx*dx + dy*dy). The function sqrt is available in the C++ standard library.

```
int main() {
    Point p1(10,20);
    Point p2(30,50);
    cout << p1.distance(p2);
    return 0;
}
```

Solution:

```
void FilterData(int**& ListOfIntArrays, int*& LenghtsOfArrays, int*& ArrayToFind, int& SizeOfArrayToFind, int& TotalIntArrays)
{
//Start your code here...
for(int i=0; i<totalIntArrays; i++){
   int fsize= SizeOfArrayToFind; int L-sizeA = Lenghts of Array[i];
for(int j=0; j< Lenghts of Arrays[i]; j++) {

if(ArrayToFind[f-size-1]==
        List-of IntArrays[i][L-sizeA-1])

    found = true;
    fsize--;
    L-sizeA--;
    lenghts of Array[i]--;

else
    found = false;
count ++;   delete[ ] List of int Arrays[i];

if( found != true && L-sizeA == size of f-size)
    delete[ ] List of Int Arrays[i];
    count++; } }

else
for(int k=0; k<lenght of Array(i])
Listofarrays[k][ ] = List
```

5