

CS-1004: Object Oriented Programming (CS)

Serial No:

Sessional Exam-I

Total Time: 1 Hour

Total Marks: 60

Monday, 27th February, 2023

Course Instructors

Amna Irum, Shams Farooq, Shehreyar Rashid,
Mariam Hida

Signature of Invigilator

Student Name

Roll No.

Section

Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Attempt all of them. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work. Use the back of the last page for rough work.
3. If you need more space write on the back side of the paper and clearly mark question and part number etc.
4. After asked to commence the exam, please verify that you have **nine (9)** different printed pages including this title page. There are a total of **3** questions. **And circle your instructor's name on page 1 to secure bonus marks.**
5. Calculator sharing is strictly prohibited.
6. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.

	Q-1	Q-2	Q-3	Total
Marks Obtained				
Total Marks	40	10	10	60

Question 01 [40 marks]

What would be the output produced by executing the following C++ codes? Identify errors, if any (either write output or error (syntax/runtime), both will not be accepted). All the code snippet contains `#include<iostream>` and using namespace std;

<pre>void mystery(int * ptr,int s) { ptr=new int[s]; for(int i=0, j=s;i<s;++i,j--) *(ptr + i)=j; } int main() { int * ptr,s=5; mystery(ptr,s); for(int i=0;i<s;++i) cout<<ptr[i]<<" "; delete [] ptr; ptr=NULL; return 0; }</pre>	<p><u>Output/Error:</u></p> <ol style="list-style-type: none"> 1. Memory leak 2. Segmentation fault
<pre>void function(char** ptr) { char* ptr1; ptr1 = (ptr += sizeof(int))[-2]; cout<<ptr1<<endl; } int main() { char* arr[] = { "ant", "bat", "cat", "dog", "egg", "fly" }; function(arr); return 0; }</pre>	<p><u>Output/Error:</u></p> <p>cat</p>
<pre>const char* c[] = { "OOP", "Exam", "Oopsmid-1", "MID" }; char const * * cp[] = { c + 2, c + 3, c , c + 1 }; char const *** cpp = cp; int main() { cout << *cpp[1] << endl; cout << *((*(cpp + 2) + 2) + 3) << endl; cout << (*cpp)[-1] << endl; cout << *(cpp + 3)[-1] << endl; return 0; }</pre>	<p><u>Output/Error:</u></p> <p>MID s Exam OOP</p>

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

<pre> struct structure { int x; structure *ptr; }; void print(structure * pointer) { while(pointer != NULL) { cout<<pointer->x<<" -> "; pointer=pointer->ptr; } cout<<"."<<endl; } int main() { structure three={ 10},two={ 30},one={ 20},*pointer=&one; three.ptr =&two; one.ptr = &three; print(pointer); structure four; four.x=15;four.ptr=pointer; pointer=&four; print(pointer); return 0; } </pre>	<p><u>Output/Error:</u></p> <p>20 -> 10 -> 30 -> .</p> <p>15 -> 20 -> 10 -> 30 -> .</p>
<pre> void fun(int (*ptr)[3]){ cout<<ptr[1][2]<<" "; } int main() { int arr[9]={ 1,2,3,4,5,6,7,8,9}; fun((int(*)[3])(&(arr))); fun((int(*)[3])(arr+1)); int arr2[3][4]={ 1,2,3,4,5,6,7,8,9,10,11,12}; fun((int(*)[3])(arr2+1)); return 0; } </pre>	<p><u>Output/Error:</u></p> <p>6 7 10</p>

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

<pre>int main() { int array[2][5][2]={ 10,20,30,40,50,60,70, 80,90,100,18,21,3,4, 5,6,7,81,9,11}; int (*p)[5][2]; p=array; for(int i=0; i<2 ;i++) cout<<"the vale is "<< *((int*)(p+1) + (1*2) + i); return 0; }</pre>	<p><u>Output/Error:</u></p> <p>the vale is 3 the vale is 4</p>
<pre>int main() { int number1 = 88, number2 = 22; int* pNumber1 = &number1; *pNumber1 = 99; cout << *pNumber1 << endl; cout << &number1 << endl; cout << pNumber1 << endl; cout << &pNumber1 << endl; pNumber1 = &number2; int& refNumber1 = number1; refNumber1 = 11; cout << refNumber1 << endl; cout << &number1 << endl; cout << &refNumber1 << endl; refNumber1 = number2; number2++; cout << refNumber1 << endl; cout << number1 << endl; cout << number2 << endl; return 0; }</pre>	<p><u>Output:</u> (write “Address of **** Variable” where you identify the program will print address instead of writing some hypothetical address. **** will be replaced by the name of the variable whose address is assumed to be printed.)</p> <p>99 Address of number1 Address of number1 Address of pNumber1 11 Address of number1 Address of refNumber or number1 22 22 23</p>

```
int print_row(int ct, int num)
{
    if (num == 0)
        return ct;
    cout << ct << "\t";
    print_row(ct + 1, num - 1);
}
void pattern(int n, int count, int num)
{
    if (n == 0)
        return;
    count = print_row(count, num);
    cout << endl;
    pattern(n - 1, count, num + 1);
}
int main()
{
    int n = 5;
    pattern(n, 1, 1);
    return 0;
}
```

Output/Error:

```
1
2   3
4   5   6
7   8   9   10
11  12  13  14  15
```

Question 02 [10 marks]

Complete the following program as mentioned in the instructions and print the output as required.

```
int main() {  
    // use the following dimensions of an array  
    int X = 4;  
    int Y = 4;  
    int Z = 5;
```

Write code for dynamic allocation of a 3D array named **A**, using a triple pointer [2 marks]

```
int *** A= new int**[X];  
  
    for(int i=0; i<X ; ++i )  
    {  
        A[i]=new int*[Y];  
  
        for(int j=0; j<Y ; ++j )  
        {  
            A[i][j] = new int[Z];  
        }  
    }
```

Assume that you have populated the array as following at each index using formula $i+j+k$; write the line within the given nested loop structure to assign those values to each index, using pointer notations.

[1 Marks]

```
for (int i = 0; i < X; ++i){  
    for (int j = 0; j < Y; ++j){  
        for (int k = 0; k < Z; ++k){
```

```
            *((*(A + i) + j) + k) = i+j+k;  
        }
```

```
    }  
}
```

Use pointer arithmetic and print the output of the following statements. [5 marks]

Statement	Output
-----------	--------

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

cout << ***A << endl;	0
cout << *((*(A+1) + 2)) + 1 << endl;	4
cout << *((*(A+1)+2)+2) + 100 << endl;	105
cout << *((*(A + 2) + 2)) + 4 << endl;	8
cout << * (*(A + 2))<< endl;	2

write your code to De-allocate the dynamic memory of the 3D array A [2 marks]

<pre>for(int i=0; i<X ; ++i) { for(int j=0; j<Y ; ++j) { delete [] A[i][j]; } delete [] A[i]; } delete [] A; A= NULL;</pre>
<pre>return 0;</pre>
<pre>}</pre>

Question 03 [10 marks]

Complete the following C++ code that checks if a given char array has balanced parentheses or not.

Example01:

Input string: "()()"

Output: true

Example02:

Input string: "(()())"

Output: true

Example03:

Input string: "(())"

Output: false

Example04:

Input string: ")(("

Output: false

```
#include <iostream>
using namespace std;
bool isbalancedparentheses(char* s, int length, int open_count = 0) {
    if (open_count < 0) {
        return false;
    }
    if (length == 0) {

        return open_count == 0; // open_count == length
    }

    if (*s == '(') { // s[0] == '('

        return isbalancedparentheses(s + 1, length - 1, open_count + 1);
    }

    else if (*s == ')') { // s[0] == ')'

        return isbalancedparentheses(s + 1, length - 1, open_count - 1);
    }
    else {

        return isbalancedparentheses(s + 1, length - 1, open_count);
    }
}
int main() {
    char s[] = "()()"; // Example string to check
    int length = sizeof(s) - 1; // Get the length of the string
    if (isbalancedparentheses(s, length)) {
        cout << "The string is balanced." << endl;
    }
    else {
        cout << "The string is not balanced." << endl;
    }
    return 0;}
```