National University of Computer and Emerging Sciences

FAST School of Computing **Spring-2024** **Islamabad Campus**

CS-1004: Object Oriented Programming (CS)

Serial No:

1st Sessional Exam

Total Time: 1 Hour
Total Marks: 60

Course Instructors

_____
Signature of Invigilator

_____	_____	_____	_____
Student Name	Roll No.	Course Section	Student Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work.
3. Verify that you have **eight (8)** different printed pages including this title page. There are **two (2)** questions.
4. Calculator sharing is strictly prohibited.
5. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.
6. Please consider g++ compiler for all questions in this paper.
7. Ensure that you do not have any electronic gadget (like mobile phone, smart watch, etc.) with you.

	Q-1	Q-2	Total
Marks Obtained			
Total Marks	**50**	**10**	**60**

Question 1 [50 Marks]

Write the output of the following C++ codes (if the code is correct). If you find any error/s in the code, please identify and explain the error/s (Note: *do not write output if there is an error*). Assume that required libraries and main function are already included in the program.

i. [2 Marks]

```
bool x = 0;
int y = 19;
char z = 's';
void* ptr = &z;
int* i = (int*)ptr;
char* c = &y;
cout << *ptr;
```

Output/Error:

ii. [2 Marks]

```
int var1 = 170;
int *p= &var1;
const int* ptr = p;
cout<< *p <<" "<< *ptr <<endl;
cout<< (*p)++ <<" "<< ++(*ptr);
```

Output/Error:

iii. [2 Marks]

```
char *s[ ] = {"black", "white", "yellow", "violet"};
char **ptr[ ] = {s+3, s+2, s+1, s}, ***p;
p = ptr;
***p='a';
cout<<**p<<endl;
```

Output/Error:

iv. [6 Marks]

```
char *s[ ] = {"black", "white", "yellow", "violet"};
char **ptr[ ] = {s+3, s+2, s+1, s}, ***p;
p = ptr;
cout<<**(++p)<<endl;
cout<<*(--(*(++p))) + 3<<endl;
```

Output/Error:

v. [6 Marks]

```
struct IntArray{
int * arr,size;
void create(int *ptr,int s){
    size=s;
    arr=ptr;
}
void display(){
    for(int i=0;i<size;i++)
        cout<<arr[i]<<" ";
    cout<<endl;
}};
int main(){
    int arr[]={4,0,3,1,2};
    IntArray my_arr;
    my_arr.create(arr+2,3);
    arr[my_arr.arr[0]]=arr[arr[1]];
    my_arr.display();
    return 0;
}
```

Output/Error:

vi. [6 Marks]

```
void fun(int (*p)[3]){
    cout<<p[-1][-4]<<endl;
}
void fun(int (*p)[2][3]){
   static bool flag=true;
   if(flag){
    cout<<p[-2][0][7]<<endl;
    flag=false;
    fun(p - 1);
   }
    fun((int(*)[3])p);
}
int main()
{
int ary[6][3] = {{1, 2, 3},{7,8,9},{4,5,6},{10,11,12}};

int (*ptr)[4]=(int(*)[4])(ary+2);

ptr--;

    fun((int(*)[2][3])(&ptr[2][5]));
return 0;
}
```

Output/Error:

vii. [6 Marks]

```
class integer{
    private:
        int i;
    public:
        integer(int ii){i=ii;}
        int getI(){return i;}
        void setI(int ii){i=ii;}
};
void display(integer i){
    cout<<"integer is "<<i.getI()<<endl;
}
void decrement(integer i){
          i.setI(i.getI()-1);}

void increment_decrement(integer & i){
    static int s;
    if(s == 0){
        i.setI(i.getI()+1);
        s++;
    }
    else{
        decrement(i);
        s--;
    }
    display(i);
}
int main(){
    integer i(10);
    display(i);
    increment_decrement(i);
    increment_decrement(i);
    increment_decrement(i);
return 0;
}
```

Output/Error:

viii. [5 Marks]

```
int list[5]={2,4,8,10,-1};
int nextList[5]={3,-1,0,1,-1};
int start = 2;
int Free = 4;
void magic(int val , int position){
    int start = ::start;
    for(int i = 0 ; i< position - 1 ; i++)
        start=nextList[start];
    list[Free]=val; nextList[Free]=nextList[start];
    nextList[start]=Free++;
}
void magic(){
    int start = ::start;
    while(start != -1){
        cout<<list[start]<<"->";
        start=nextList[start];
    }
    cout<<"*"<<endl;
}
int main(){
    magic();
    magic(5,2);
    magic();
    return 0;
}
```

Output/Error:

ix. [15 Marks]

```
struct Box1 {
  int x;
  Box1 *ptr1;
};
struct Box2 {
  Box1 *ptr1;
  Box2 *ptr2;
};
void print(Box2 * ptr1){
  while(ptr1){
      Box1*  ptr2=ptr1->ptr1;
    while(ptr2){
      cout<<ptr2->x<<" ";
      ptr2=ptr2->ptr1;
    }
    cout<<endl;
    ptr1=ptr1->ptr2;
  }
}
int main(){
    Box1 three={10},two={30},one={20},*pointer=&one;
    pointer->ptr1 = &three; (*(*pointer).ptr1).ptr1=&two;
    Box2 one1={&two},two1={pointer} ,
three1 = {(*pointer).ptr1,&one1} , *pointer2=&three1;
   (*(*pointer2).ptr2).ptr2=&two1;
   print(pointer2);
return 0;
}
```

Output/Error:

Question 2 [10 Marks]

Complete the following recursive function (C++) which finds a substring in a given string. Analyze the code and write the missing statements to make the code executable.

Example 01	Example 02:	Example 03
text =“OOPquizspring” pat = “quiz” output : true	text =“OOPquizspring” pat = “quiz1” output : false	text =“OOPquizspring” pat = “OOPquizspring” output : true

```
#include<iostream>
using namespace std;
bool exactMatch(char *text, char *pat){


    if (_____________________________________________)     //[2 marks]
        return false;

    if (*pat == '\0')
        return true;
    if (*text == *pat)


        return ____________________________________;        //[2 marks]


    return_____________________________; //            [1 marks]
}

bool contains(char *text, char *pat){

    if (*text == '\0')

        return ____________________________________;          //[1 marks]

    if (*text == *pat)

        if(_____________________________________________) //[2 marks]
            return 1;
        else

return contains(_________________,_____________________);//[2 marks]

    return contains(text + 1, pat);
}
int main(){
 cout << contains("OOPquizspring", "quiz") << endl;
 return 0;
}
```