National University of Computer and Emerging Sciences

Lahore Campus

Object Oriented Programming (CS1004)

Sessional-II Exam

Total Time (Hrs): 1
Total Marks: 40
Total Questions: 2

Date: April 06, 2024

Course Instructor(s)

Mr. Aamir Rahim
Ms. Anosha Khan
Ms. Arooj Khalil
Ms. Samin Iftikhar
Mr. Uzair Naqvi
Mr. Waqas Manzoor

[redacted] Roll No

[redacted] Section

[redacted] Student Signature

IMPORTANT INSTRUCTIONS: Do not use pencil or red ink to answer the questions. In case of confusion or ambiguity make a reasonable assumption. **Only one solution will be marked against one question, carefully attempt questions on answer sheet.**

CLO # 1: Demonstrate the basic concepts of OOP

Q1: [10+10 = 20 marks] Short Questions: Write output (or error, if any) of the code segment below. In case of error clearly mention the error. (There is no syntax error in the code.)

Part (a) [10 Marks]

```
#include <iostream>
#include <cstring>
using namespace std;
class Person {
private:
    char* name;
public:
    Person(const char* n) {
        name = new char[strlen(n) + 1];
        strcpy(name, n);
              cout<<"Person() Called.\n";
    }
    ~Person() {
              if(name) delete[] name;
    }
    void displayName(){
        cout << "Name: " << name << endl;
    }
       Person(Person& rhs){
              cout<<"Copy Person Called.\n";
              name = rhs.name;
       }
       Person& operator=(Person& rhs){
              cout<<"Op = Called.\n";
              name = rhs.name;
       }
};
int main() {
    Person person1("Alice");
    Person person2 = person1;
    {
        Person person3 = person1;
             cout << "Person 3: ";
             person3.displayName();
             cout<<"------------\n";//just
draw a small line
    } → Destruct P3
    cout << "Person 1: ";
    person1.displayName();
    cout << "Person 2: ";
    person2.displayName();
    return 0;
}
```

Part (b) [10 Marks]

```
#include <iostream>
using namespace std;
class Box {
       int* length;
public:
       Box(){
              length = new int;
              *length = 99;
       }
       Box(Box& rhs){
              length = new int;
              *length = *rhs.length;
       }
       Box& SomeFunction(int l){
              *length = l;
              return (*this);
       }
       ~Box(){
              cout << "Box destroyed. Length
was: " << *length << endl;
              if(length != 0)
                     delete length;
       }
       void Print(){
              cout<<*length<<endl;
       }
       void AnotherFunction()      {
              Box temp;
              *temp.length = 555;
              cout<<"Printing... "; Print();
              *this = temp;
       }
};
void main() {
       {
              Box obj1;
              {
                     Box* obj2 =
&(obj1.SomeFunction(10));
                     cout<<"Obj 2 = ";
                     obj2->Print();
              }
              cout<<"Obj 1 = ";
              obj1.Print();
       }
       cout<<"------------\n"; //just draw a
small line
       {
              Box obj1;
              obj1.AnotherFunction();
              cout<<"Obj 1 = ";
              obj1.Print();
       }
}
```

CLO # 2: Apply OOP concepts (Encapsulation, Inheritance, Polymorphism, Abstraction) to computing problems for the related program

Q2: [20 marks]

You are required to implement a class **MyList** that manages a dynamically growing array of strings. Partial class definition, driver program (main) and required output is given below. **Implement the class MyList such that the given main program runs successfully.** *Make sure that your program doesn't consume extra space and there shouldn't be any memory leakage or exceptions in your code.*

Note: *You can use built-in string functionality like strlen, strcpy etc. where required. You do not need to re-write these functions.*

Partial Definition of class and Main Program:

```
class MyList{
	//You cannot add or modify DATA members
	char** arr; //Array of Strings
	int size;	//No of Strings in List
};
void main()
{
	MyList list1; //Initially list will be empty.
	cout<<"list1 = "<<list1;	//Displays all the strings in list

	"Apple" + list1; //Adds new string "Apple" in list1
	cout<<"list1 = "<<list1;	//Displays all the strings in list

	"Banana" + list1; //Adds new string "Banana" in list1
	cout<<"list1 = "<<list1;

	"Peach" + list1; //Adds new string "Peach" in list1
	cout<<"list1 = "<<list1;

	MyList list2; //Initially list will be empty.
	cout<<"list2 = "<<list2;

	"I love Pakistan" + list2; //Adds new string "I love Pakistan" in list2
	cout<<"list2 = "<<list2;

	"Happy Programming" + list2; //Adds new string "Happy Programming" in list2
	cout<<"list2 = "<<list2;

	MyList list3 = list1+list2; //Creates another object of MyList having all the strings
of list 1 and 2
	cout<<"List 3 = "<<list3;
}
```

Required Output:

```
list1 = []
list1 = [Apple]
list1 = [Apple, Banana]
list1 = [Apple, Banana, Peach]
list2 = []
list2 = [I love Pakistan]
list2 = [I love Pakistan, Happy Programming]
List 3 = [Apple, Banana, Peach, I love Pakistan, Happy Programming]
```