



NATIONAL UNIVERSITY
OF COMPUTER & EMERGING SCIENCES
LAHORE



Course Object Oriented Programming Answer Sheet No. 12071
Student's Name Muhammad Saboor Islam Signature Saboor
Roll No. 231 3028 Section CS 2D 2 Date _____

Question No 02:

```

class MyList {
private:
    char ** arr;
    int size;
public:
    MyList() {
        arr = nullptr;
        size = 0;
    }
    friend operator + (char [] a, MyList list) {
        char ** temp = new char [list.size + 1];
        for (int i = 0; i < list.size; i++) {
            temp[i] = new char [strlen(arr[i])];
            strcpy(temp[i], arr[i]); // src, dest, src
        }
        list.temp[size] = new char [strlen(a) + 1];
        strcpy(temp[size], a);
        for (int i = 0; i < list.size; i++) {
            delete list.arr[i];
        }
        delete [] list.arr;
        list.size++;
        list.arr = new char [list.size];
        for (int i = 0; i < list.size; i++) {
            list.arr[i] = new char [strlen(temp[i])];
            strcpy(list.arr[i], temp[i]);
            delete [] temp[i];
        }
        delete [] temp;
    }
}
  
```



```

• friend ostream& ostream& print (
    MyList list) &
    print << "[ ";
    for (int i=0; i < list.size; i++) &
        print << list.arr[i];
    print << " ";
    }
    print << "] ";
    return print;
    }

```

```

• ~ MyList() &
    for (int i=0; i < size; i++) &
        delete [] arr[i];
    }
    delete [] arr;
    arr = nullptr;
    size = 0;
    }

```

```

• MyList& operator + (const MyList& rhs) &
    MyList temp;
    for (int i=0; i < size; i++) &
        arr[i] + temp;
    }
    for (int i=0; i < rhs.size; i++) &
        arr[i] + temp;
    }
    return * (temp); // value (scope)
    }
    }

```

Copy

(10)

Question NO 01

(a)

~~Person()~~ called. ~~Copy Person~~ called.
Copy Person called. ~~Person B: Name: Alice~~

~~Person 1:~~ Error

↳ Trying to access pointer that is deleted when destructor called for Person 3.

~~Person 2:~~ Error

↳ Again Trying to access pointer already deleted due to end of scope of Person 3. We should use deep copy instead of shallow copy.

(b)

~~Obj 2 = 10~~ → Box destroyed. Length was 10

Obj 1 = Error

↳ Obj 2 pointing to Obj 1. when scope of Obj 2 ended it destroyed length with it so dangling pointer

Box destroyed. Length was Error

↳ Same Error.

~~Printing ...~~ 99

~~Obj 1 =~~ Error 555

Box destroyed. Length was 555