15

MIDTERM EXAM ANSWER SHEET

NATIONAL UNIVERSITY
OF COMPUTER & EMERGING SCIENCES
LAHORE

FAST

|  |  |  |  |
|---|---|---|---|
| Course | OOP | Answer Sheet No. | 40387 |
| Student's Name | Faiq Saeed | Signature | [illegible] |
| Roll No. 23L-0905 | Section BCS-2B | Date | April 6, 2024 |

31

31 → 1.9

## Question#01

a) Person Called
Copy Person Called
Copy Person Called
Person 3: Name: Alice
- - - - - - - - - -
Person 1: Name: Alice
Person 2: Name: Alice

(logical error that shallow copy is created even tho after deleting it once it can still be accessed but it's not a good programing practice)

b) ~~Que~~

10

Obj2 = 10
Box destroyed. Length was 10
Obj1 = 10
Box destroyed. Length was 10
— — — — — — — —
Printing.... 99
Box destroyed. Length was 555
555
Box destroyed. Length was 555

rt No. Rough Work

## Question#02

12

```
class MyList{

    char** arr;
    int size;

public:
friend ostream& operator<<(ostream&, MyList&);
  ~~MyList (int s=0 const char** a=nullptr)~~
  ~~{ arr = a;~~
  ~~  size = s;~~
  ~~  if(s){ arr = new char* [s];}~~
  ~~      for(int i=0; i<~~
  ~~}~~
  MyList( Mylist& RHS                    )
  {

    this->size = RHS. size;
    ~~for(int ( i=0 ; i<this->size ; i++)~~
    ~~{~~
    ~~this->char = new~~
    char  = new char*[size];
    for (int i=0; i< size ; i++)
    {~~int this->arr[0]~~
         int l= size of (RHS.arr[i][ ])/size of(RHS[i][0])
         char [i] = new char[l];
    {    for(int  j=0 ; RHS.char[i][j] != "\0"; j++)
         {
            this->char[i] [j] = RHS.char [i][j];
         }
    }
  }       // copy constructor;

  MyList(int s =0 ; const char** a =nullptr)
  {
   size =s
   arr =a
  if (size !=0)
  {    char = new  char*[size];
       //Next page
```

Q / Part No.

```
 a + b -> abc
MyList operator+(MyList &(abc)
{ ~~MyList temp;~~
  int length = this-> size + abc.size;


    for(int i=0; i< size; i++)
    {
       int  r = size of (a[i][])/size of(a[i][0]);
       char [i] = new char [r];
       for(int j=0; a[i][j] != "\0"; j++)
       {
          char[i][j] = a[i][j];
       }
    }
}// constructor default+own


MyList operator+(MyList & abc)
{ if( this != &abc)
 { int  a = this-> size + abc. size;
   charr ~~**~~ arrc;
   arrc = new char* [a]
   ~~fr~~
   int  i = 0;
   for(    ; i<this->size; i++)
   {   int  r = size of(this->~~char~~[i][])/sizeof(this->[i][0]
       arrc
       ~~char~~[i] = new char [r];
     for (int j =0; this->char[i][j] != "\0"; j++)
     {
           arrc[i][j] = this->arr[i][j];
     }
   } ~~i++;~~
 for(int  j=0; j< abc.size ;i++, j++)
 { int ~~a~~ = d[sizeof(abc.arr[i][])/sizeof(abc[i]
                                                 [0]
```

4

art No. Rough Work

```
        arrc[i] = new char[n];
        for(int k = 0 ; abc[j][k] != "\0"; k++)
        {
            arrc[i][k] = abc.arr[j][k]
        }

    MyList temp(a, arrc);
    return temp;} //if check ends
}
```

5

```
MyList  operator+(const char* a)
{
      int l = ++this->size;
      char** temp1 = new char*[l]; int i = 0;
      for(~~int i~~ = 0;  i < this->size ; i++)
      {
         int n = size of(this->~~char~~[i][ ])/size of(this->~~abc~~[i][0])
         temp1[i] = new char[n];
         for(int j = 0 ; this->arr[i][j] != "\0"; j++)
         {
            temp1[i][j] = this->arr[i][j];
         }
      }
      int n = size of(a[ ]) /size of (a[0]);
      ~~for(int j = 0;~~
       temp1[i] = new char[n];
      for(int j = 0 ;  a[j] != "\0"  j++)
      {    ~~char~~
            temp1[i][j] =  a[j];
      }
      MyList temp(l, temp1)
      return temp;
}
```

```
~MyList()
{ if(arr)
  {for(int i=0 ; i< size; i++)
    { delete[] arr[i];}
    delete [] arr;
    arr = null ptr;
  }

};

ostream& operator <<(ostream& out, MyList& a)
{

    ~~cout~~
    cout<<"[";
   ~~out~~
   for(int i=0 ; i< a.size ; i++)
   { ~~out <~~
      for(int j=0 ; g.char[i][j] != "\0" ; j++)
      {
         out << a.Char [i] [j];
      }
    cout<<",";
   }
  cout << "]";

 return out;
}
```