National University of Computer and Emerging Sciences

School of Computing **Spring 2015** **Islamabad Campus**

CS103
Computer Programming

Tuesday, February 16, 2016

Serial No:

Sessional I

Total Time: 1 Hour

Total Marks: 88

Course Instructor(s)

Sibt ul Hussain, Usman Farrokh and Hassan Mustafa

_____

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 (**9**) different printed pages including this title page. There are total of (**4**) questions.
5. Use of calculator 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.
7. Use **proper indentation** while writing code and make sure that your code is legible. Failing to do so can cost you marks.
8. **Please allocate your time properly according to the marks distribution.**
9. Write proper explanation of the error (or bug) where required, without proper explanation no marks will be awarded.

	I	II	III	IV	**Total**
Total Marks	46	10	12	20	88
Marks Obtained					

Vetted By: _____ Vetter Signature: _____

Question I . . . . . **(46 Marks)**

Please write proper explanation of the error (or bug) where required, without proper explanation no marks will be awarded. **Note: there are no syntax errors, whatsoever.**

(a) **(1 Mark)** Define an alias to a pointer to an integer.

(b) **(2 Marks)** Define an variable to store the address of following variable.

```
char **a[4];
      = &a; // write the valid expression to store the address of a
```

(c) **(2 Marks)** Write the code to deallocate following memory

```
char *a[4];
char n[90]=''hello''
a[0]=a[3]=n;
a[1]=new char [40];
a[2]=new char [60];
```

(d) **(4 Marks)** Write the code to dynamically allocate memory for a 2-dimensional array (take number of rows and columns input from the user) and then deallocate the allocated memory.

(e) **(5 Marks)** What will be the output of following code. Explain the error or bug if there is any.

```
#include <iostream>
using namespace std;
int main(){
        int m[30];
        for (int i = 0; i < 30; ++i)
                m[i]=i+1;
        int **q=new int*[5];
        for (int i = 0; i < 5; ++i)
                q[i]=&m[i*6];
```

```
10          for (int i = 4; i >=0 ; --i)
11          {
12                  for(int j=0; j <6;++j)
13                          cout<<q[i][j]<<" ";
14                  cout<<endl;
15          }
16          delete []q;
17          return 0;
18  }
```

(f) **(5 Marks)** What will be the output of following code. Explain the error or bug if there is any.

```
1   #include <iostream>
2   using namespace std;
3   int *ComputeTable(int indeces[], int number)
4   {
5           int result[10];
6           for (int i = 0; i < 10; ++i)
7           {
8                   result[i]=number*indeces[i];
9           }
10          return result;
11  }
12  int main(){
13          int num[]={1,2,3,4,5,6,7,8,9,10};
14          int *p=ComputeTable(num,5);
15          for (int i = 0; i < 10; ++i)
16                  cout<<p[i]<<" ";
17  }
```

(g) **(7 Marks)** What will be the output of following code. You will have to draw the calling stack as well.

```
1   #include <iostream>
2   using namespace std;
3
4   int CTD(int, int, int);
5   void main()
6   {
7           int n=12343, b=5;
8           cout<<CTD(n, 1, b)<<endl;
9   }
```

```
int CTD(int n, int t, int b)
{
        if(n > 0)
        {
                return (n % 10)*t + CTD(n/10, t*b, b);
        }
        return 0;
}
```

(h) **(5 Marks)** What will be the output of following code. Explain the error or bug if there is any.

```
#include <iostream>
#include<cstring>
using namespace std;
void Mystery(char*dest, char *src){
        int i=0;
        int e=strlen(src);
        while(i < e)
        {
                *dest=*src;
                dest++;
                src++;
                i++;
        }
        *dest='\0';
        cout<<dest;
}
int main( ){
        char *src="ABC";
        char *dest="DEF";
        Mystery(dest,src);
        cout<<dest;
        return 0;
}
```

(i) **(5 Marks)** What will be the output of following code.

```
#include <iostream>
#include <cstring>
using namespace std;
char* Mystery(char *& rptr)
```

```
{
        char *n=new char[6];
        char *str2 = "World!";
        strcpy(rptr,str2);
        strcpy(n,"Hello");
        return n;
}
int main () {
        char *str1=new char[6];
        strcpy(str1,"Hello");
        cout <<Mystery(str1) << " " <<str1;
}
```

(j) **(5 Marks)** What would be the output produced by executing the following C++ code segment?

```
#include <iostream>
using namespace std;
struct Puzzle
{
        int x, y;
        void Init(int x1, int y1)
                {
                        x=x1;
                        y=y1;

                        int& b = x;
                        int* c = &b;
                        cout << &x << " ";
                        cout << &b << " ";
                        cout << &(*c) << endl;

                        int& d = y;
                        int* e = &d;
                        cout << &y << " ";
                        cout << &d << " ";
                        cout << &(*e) << endl;

                }
};

int main() {
        Puzzle p;
        p.Init(4,5);
        return 0;
}
```

(k) **(5 Marks)** What would be the output produced by executing the following C++ code segment?

```
#include <iostream>
#include <cstring>
using namespace std;
// Given the following structure
struct A
{
        int * x;
        int z;
        void Init()
        {
                x= new int;
                *x=20;
                z=10;
        }
        void Print()
        {
                cout <<*x <<" "<<z <<endl;
        }
};
struct B{
        A t;
        int a;
        void Init(A & obj, int a1)
        {
                t=obj;
                *t.x=40;
                t.z=32;
                a=a1;
        }
        void Print()
        {
                t.Print();
                cout << a <<endl;
        }
};
int main()
{
        A a;
        a.Init();
        B b;
        b.Init(a,5);
        b.Print();
        a.Print();
        return 0;
}
```

Question II . . . . . (10 Marks)

Write a recursive function to reverse elements of a given array. For example given following array as input

```
int␣a[]={3,4,9,8,5,7}
```

your function must update the array to contain following contents

```
{7,5,8,9,4,3}
```

Your function must work for both even and odd number of elements. Your function can at most receive 4 arguments, and you are not allowed to use global or static variables.

Question III ..... **(12 Marks)**

Write a function `findsubstr` that finds the locations of all the instances of a substring in a given string. The function should receive 2 `char*` variables as arguments *i.e.* a string, and another string which is to be found in the first string. It should return an integer array with the starting positions of all the found substrings instances. The first element of the returned array should be the number of instances found.

For example, the call `findsubstr("Sue Queue","ue")` should return [3, 1, 5, 7]. If no instance of the substring is found, your function must return NULL.

Note: You cannot use string class for this task.

```
1   int * findsubstr(char *, char*);
```

Question IV . . . . . (20 Marks)

Your goal in this question is to write a program to manage a Parking Garage. For this you will need to create a structure Parking Garage that should be able to simulate a basic parking garage with the following functionalities:

Your garage should have a fixed capacity (5 cars at most). For each car stationed at your garage you will record its entry time (current day hour in 24h format), its registration number and its allocated slot (each car can be allocated parking slot 1 to 5). A new car cannot be parked if the garage capacity is already full.

Whenever a new car arrives at your garage you will add it to the garage if the capacity is not full. When a car leaves the garage you will ask the user current time (hour in 24 hour format) and charge him Rs. 20 per hour, and declare the slot as available so that a new car can be parked here.

Define and use a structure (or class) for the Garage. Identify all the data members and member functions of this structure (or class) and write them. For instance, your class or structure must provide, apart from other functions, following interface (`public`) functions.

Member Function	**Description**
Initialize	Sets all the member to reasonable values.
Print	Should print the currently parked cars information and empty slots information
IsFull	Should return whether Garage is full or not.
ParkCar	Should add a new car to the Garage if it is not full.
RemoveCar	Should remove a car from a given parking slot.