0088

National University of Computer and Emerging Sciences

FAST School of Computing Fall-2023 Islamabad Campus

# CS-1002: Programming Fundamentals (CS)

Wednesday, 27th December, 2023

Serial No:

**Final Exam**
**Total Time: 3 Hour**
**Total Marks: 170**

## Course Instructors

Dr. Muhammad Arshad Islam, Mr. Shehryar Rashid,
Mr. Aqib Rehman

**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 **Eighteen (18)** different printed pages including this title page. There are total of 6 questions.
5. Calculator sharing 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.

| | Q-1 | Q-2 | Q-3 | Q-4 | Q-5 | Q-6 | Total |
|---|---|---|---|---|---|---|---|
| **Marks Obtained** | 2.5 | 5 | 05 | ~~20~~ | 9.5 | 8.5 | 48 |
| **Total Marks** | 5 | 10 | 15 | 45<br>18 | 50 | 45 | 170 |

0L

# National University of Computer and Emerging Sciences

FAST School of Computing   Fall-2023   Islamabad Campus

## Question 1 [5 Marks]

Write the output of the following C++ codes (if the code is correct). If you find any error/s in the code, identify it, correct it and then write the output. Assume that required libraries and main function are already included in the program.

Note: No marks for direct output. Show the dry run.

i. [2.5 Marks]

```
int main() {
    const int size = 5;
    int* arr = new int[size];
    for (int i = 0; i < size; ++i) {
        arr[i] = i * 2;
    }
    int* ptr = arr + 2;
    cout << *ptr << endl;
    delete[] arr;
    return 0;
}
```

Dry run:

arr[0]

arr[1] = 1×2 ; 2
arr[2] = 2×2 4 arr+2
arr[3] = 3×2 6
arr[4] = 4×2 8
arr

*(arr+2)

Output:

4   2.5

ii. [2.5 Marks]

```
void allocateMemory(int** ptr) {
    *ptr = new int;
    **ptr = 42;
}
int main() {
    int* value = nullptr;
    allocateMemory(&value);
    cout << &value<<" "<<value<<" "<<*value<< endl;
    delete value;
    value= nullptr;
    return 0;
}
```

Output:

0088

## National University of Computer and Emerging Sciences

FAST School of Computing Fall 2011 Islamabad Campus

### Question 2 [10 Marks]

Write the output of the following C++ codes (if the code is correct). If you find any error in the code, identify it, correct it and then write the output. Assume that required libraries and main function are already included in the program.

Note: No marks for direct output. Show the dry run.

i. [4 Marks]

```
#include <iostream>
using namespace std

int j=19
int* fun()
{
     static int j=1;
     if(1)
     {
          static int j=9;
          j--;
     }
     ::j++;
     cout<<j;
     return &j;
}
int main()
{
    fun();
    int *qp = fun();
    cout<<(*qp+1);
    cout<<j;
     return 0;
}
```

**Output:**

**Dry Run:**

ii. [6 Marks]

```
int main()
{
    int num[5];
    int* p;
    p = num;
    *p = 10;
    p++;
    *p = 20;
    p = &num[2];
    *p = 30;
    p = num + 3;
    *p = 40;
    p = num;
    *(p + 4) = 50;
    for (int i = 0; i < 5; i++)
        cout << num[i] << ", ";

    int a = num[0], *ptr = &a;
    char ch = 'A', &cho = ch;

    cho += a;
    *ptr += ch;
    cout << a << ", " << ch << endl;

    return 0;
}
```

**Output:**

~~10,20,30,40,50~~

10, 20, 30, 40, 50,

5

**Dry Run:**

num[5] = {10, 20, 30, 40, 50}

a = 10
*ptr = &a

Question 3 [15 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. [5 Marks]

```
void magic (int* ptr, int size) {
    ptr = new int[size+1];
    for(int i=size,j=0;i>=0;i--,j++)
        ptr[j]=i;
}
int main() {
    int* ptr = nullptr;
    int size = 5;
    magic(ptr,size);
    for(int i=0; i<size ;i++)
        cout<<*(ptr + i)<<" ";      } Error
    delete [] ptr;
    ptr = nullptr;
    return 0;
}
```

Output/Error:

~~Error : me~~ Error: ~~n~~ not ~~reffer~~ referring to ptr in function but in main, which is 'nullptr'

5

ii. [10 Marks]

```
const char* c[] = { "PF", "Exam" , "centipede", "Project" };
char const * * cp[] = { c + 2, c + 3, c , c + 1 };
char const *** cpp[] = { cp + 1 ,cp + 2 };
int main()
{
  cout << ***cpp[1] << endl;
  cout << (*cpp)[-1][0] << endl;
  cout << (*cp)[-1] << endl;
  cout << (*(cpp[1][-1]) + 3) << endl;
return 0;
}
```

Output/Error:

~~cp1 C + 2~~

## Question 4 [5 + 8 + 10 + 10 + 7 Marks]

Develop a C++ program to convert a given floating-point number representing a currency value in rupees into its English phrase representation. Your program should handle values up to one lac (100,000) rupees. Following are the guidelines for implementing the currency-to-phrase conversion in C++

1. Accept a non-negative floating-point number, with at most two decimal places, as input. If more than 2 decimal places then round off to two decimal places.
2. Output the English phrase representation of the provided currency value in rupees. For instance, represent 12345 as "twelve thousand three hundred forty-five Rupees."
3. Account for decimal values up to two decimal places. For example, represent 12345.67 as "twelve thousand three hundred forty-five Rupees and sixty-seven Paisa."

18

You are required to use string datatype and can be accessed from the standard library using the string class. A few examples for using string are given below.

```
string s1 = "string initialization", s2; // declaration and initialization
cout<<"Enter String"<<endl;
cin>>s2.
cout<<"String is: "<<s1<<endl;
cout<<"Concatenated String is "<<s1 +" "+s2<<endl; // this will output s1 followed
//by one space and then s2
```

```
#include <iostream>
#include <string>
#include <cmath>
using namespace std;
string ten[] = { "", "ten", "twenty ", "thirty ", "forty ","fifty ",   // %10
"sixty ", "seventy ", "eighty ","ninety " };                          // %100
//Other Global variable here                                          // %1000
string hundred[] = {"hundred"};                                       // %10000
                                                                      // %100000
                                                                      //
string ones[] = { "one", "two", "three", "four", "five", "six", "seven",
"eight", "nine", "ten"};
string thousand[]= {"thousand"};
```

3

```
string numToWords(int num) {
string wholeNumber = "";        int n = wholeNumber
while n
```

Past

12345 - (2345) /
10000

```
string        convertNumber(int num) {
    // Implement the conversion logic for the number part here
    // Return the English phrase
~~int wholeNumber~~
int ~~wholeNumb~~ n = num;
string wholeNumberWords = "";
int digcntr = 0;
while (n!= 0) {     //check digit number
    digcntr++;
}   n/= 10;
 int digs = digcntr;

while (n!=0) {
 int multiplier = pow (10, digcntr-2);
 if ((n - (n % multiplier)) / pow(10, digcntr-3)) == 1) {
   wholeNumberwords += ones['0'] ;
    // do for each number            }
  if (digcntr == 3)  {
   wholeNumberwords += thousand[0] ; }
  else if (digcntr == 4) {
   wholeNumberwords += (ones [9] + " " + thousand [0]); }
  else if(digcntr == 5)  {
   wholeNumberwords += (hundred [0] + " " + thousand [0]); ; }
  n = n % multiplier ;
  digcntr -- ;



}
```

```
// Function to convert the currency value to English words

string convertToWords(float currencyValue) {
// Separate the whole number and decimal part
int wholeNumber = static_cast<int>(currencyValue); // 12345

int decimalPart = static_cast<int> ((currencyValue - wholeNumber) * 100)   // 67

_______________________________________________


string wholeNumberWords = convertNumber(wholeNumber);


string decimalPartWords = convertNumber(decimalPart);


// Combine the whole number and decimal part phrases


string result = wholeNumberWords + "Rupees and
+ decimalPartWords + " Paisa";

    return result;
}


int main() {
    // Example usage:
    float inputValue = 12345.67;
    string result = convertToWords(inputValue);

    cout << "Input: " << inputValue << "\nOutput: " << result <<
endl;

    return 0;
}
```

**Question 5 [3+0.5+0.5+18+0.5+9+0.5+18=50 Marks]**

In the last summer 2022 Ahmed spend his holidays by developing a C++ application to improve his logic building skills. He developed an app, which was related to arrays and pointers concepts.

Following is the flow of his application.

**You are required to fill the provided space with the appropriate code statements.**

Put the comments to explain your logic **3 Marks**.

```
#include <iostream>
using namespace std;

int main()
{
char moto[ ] = "I LOVE c++ CODING AND I will do my best";
                                   "I", " ", "L", "O", "V";
int size = 0;
// store size of array in the variable 'size'

size= sizeof(moto)                          ; //0.5 Mark     0.5

//define the pmoto pointer to point the moto array

int *pmoto = moto;                          ;  //0.5 Mark
```

/*

Write a no. of c++ statements in the below provided space to reverse the moto array using pmoto pointer in such an order that each word will change its order.

Example: After your below logic if we print the moto array the output should be:

**best my do will I AND CODING c++ LOVE I**

No need to write a function for it.

Note: You are not allowed to define and use another array. Use of any built in method for changing the order is not allowed.

*/

```
//Write your logic here   //18 Marks
~~#include<iostream>~~
~~using namespace std;~~                      // Size = 40;
~~int main()~~
~~{ for moto~~                  int * currentaddress; // current address while looping
                               ~~currentad~~          through the multiple loops
                                     current address = ~~size~~ pmoto + 38;
for (int i=0; i<4; i++)
{   *(pmoto + ((~~sizeof(moto)~~ size) - 5)) = * pmoto; // replacing the characters
    pmoto++;  // address to next ~~replacing~~ character                 // "best"
    ~~currentaddress = pmoto; // storing a currentaddress~~
}
~~pmoto~~ *(pmoto + 1) = *(pmoto(~~sizeof(moto)~~ size) - 6) // add space
                                pmoto++;
```

```
~~for (int i = 0 ; i < 2 ; i++)~~
~~{~~
~~  *(pmoto + ((sizeof (moto) - 9)) = *pmoto ;~~
~~  pmoto++; }~~
~~*(pmoto + ((sizeof(moto) - 9)) = *pmoto ;~~
~~for (int i = 0 ; i < 2 ; i++)~~
~~*(pmoto + (sizeof(moto) +~~
   ~~for (int i=0; i<2; i++)~~                // my
     currentadress = moto + 31 ;
   for ( int i=0 ; i<2 ; i++)
{
  *(pmoto + 1) = *(current address ) ;
   pmoto++ ;
   currentaddress++ ;
 }
    *(pmoto + 1) = *(current address);       current address = moto ptr + 30
    pmoto++ ;
  for (int i=0 ; i<2 ; i++) {                 // do
 *(pmoto + 1) = *(current address) ;
   pmoto++ ;    }
```

4

After the above process now we have **moto** array which contains:
best My do will I AND CODING c++ LOVE I
Now we are interested to use that character array and on the basis of their length we are converted into number array.*/
// define an array with name **numbers** of size 10 to store the integers

int numbers[10]; ; // 0.5 Mark

0.5

/*Write the c++ statements to store the characters length in **numbers** array.
After your logic the **numbers** array must contain 4 2 2 4 -1 3 -7 3 -4 -1
**Hint:** You have noticed some no.s has –ve sign so here the logic is that if the word contain all upper case letters so –ve sign will be appear with no. otherwise no sign.
*/

// Write your logic here // 9 Marks

Zero

...hich –ve numbers are

...sort that array and store in the below pattern

// define an array with name **numbers2** and initialize it with
// -7 -4 -1 -1 2 2 3 3 4 4 values

```
int numbers2 [] = { -7, -4, -1, -1, 2, 2, 3, 3, 4, 4 }; // 0.5 Mark
```

0.5

/* You are required to write no. of c++ statements to find the square of **numbers2** array elements and store them in **result** array in sorted form.
After your logic **result** array must contain
1 1 4 4 9 9 16 16 16 49
Note: You are not allowed to use any sorting method for this (Not any built in sort method not any user define sort method. **No marks for using sorting method.**
Hint: Firstly find out the position where the resultant values will be stored. */

// Write your logic here // 18 Marks

1 1 4

```
int result [10] ;

    for (int i=0 ; i < 10 ; i++)
{
    arr [i] = arr [i] * arr [i] ;      // squared
    ~~[illegible]~~
}

~~for (int i=0; i<10; i++~~
    // array is currently  49  16  1  1  4  4  9  9  16 16
    ~~for (int i=0 ; i<[illegible], i++)~~
    ~~{ if arr [i] %.~~
    ~~arr [0] = arr [2] ;~~
    ~~arr [1] = arr [3] ;~~

    ~~for (int i=0; i<10; i++)~~
    ~~{ if (arr[i]~~
    ~~arr [2] = arr [4] ;~~
    ~~arr [3] = arr [5] ;~~
```

```
~~for (int i=0; i<10;~~
  int temp = arr [0] ;
  arr [0] = arr [2] ;
  arr [1] = temp ;
                              2
int temp 2 = arr [4] ;
  ~~arr [2] = arr~~
  arr (2) = arr [3] ;
  arr (3) = temp2 ;
int temp 3 = arr [4] ;
arr (4) = arr [6] ;
arr (5) = arr [7] ;
```

2.5

+ 1.5

## Question 6 [45 Marks]

**River: [complete the code]**

You're given a two-dimensional array (a matrix) of potentially unequal height and width containing only 0s and 1s. Each 0 represents land, and each 1 represents part of a river. A river consists of any number of 1s that are either horizontally or vertically adjacent (**but not diagonally adjacent**). The number of adjacent 1s forming a river determine its length. Each 1 has at most two adjacent 1s.

Remember that a river can twist. In other words, it doesn't have to be a straight vertical line or a straight horizontal line; it can be L-shaped (or reverse). **But river doesn't have branches**

For example, 01:

```
Height = 5 and Width = 6
matrix = [
 [1, 0, 0, 1, 0, 0],
 [1, 0, 1, 0, 0, 0],
 [0, 0, 1, 0, 0, 1],
 [1, 0, 1, 0, 0, 1],
 [1, 0, 1, 1, 0, 0],
]
Output:
There is/are 5 river/s, length of each river is given below
[2, 1, 5, 2, 2]
```

For example, 02:

```
Height = 5 and Width = 6
matrix = [
 [1, 0, 0, 1, 0, 1],
 [1, 0, 1, 0, 0, 1],
 [0, 0, 1, 0, 0, 1],
 [1, 0, 1, 0, 1, 1],
 [1, 0, 1, 1, 1, 0],
]
Output:
There is/are 4 river/s, length of each river is given below
[2, 1, 11, 2]
```

For example, 03

```
Height = 5 and Width = 6
matrix = [
 [0, 0, 0, 1, 1, 1],
 [0, 0, 1, 0, 0, 1],
 [0, 0, 1, 0, 0, 1],
 [0, 0, 1, 0, 1, 1],
 [0, 0, 1, 1, 1, 0],
]
Output:
There is/are 1 river/s, length of each river is given below
[13]
```

Note: There are no syntax errors. If you think something missing, it must appear in the blanks

```
#include <iostream>
using namespace std;
const int Height = 5; //code must be generic
const int Width = 6;
void matInitialization(int mat[][Width]);//Prototype [1 marks]

int* listOfRiverLengths(int mat[][Width] , ~~int*rlist~~ int size );

 int*  insertingRiver(int * rList  , int size , int rlength);

int lengthOfRiver (int mat[][Width], int x, int y);

int main() {
    int mat[Height][Width]={{0}};
    matInitialization(mat);
    int size = 0;
    int * rList=listOfRiverLengths( mat, size );
cout<<"There is/are "<< size    <<" river/s, length of each river is
given below"<<endl;
    for( int i=0; i<size; i++)
        cout<<rList[i]<<" ";


    delete [] rList;
    return 0;
// end function
```

```
/* Function input 0s (land) and 1s (river) */
void matInitialization(int mat[][Width]){
    for( int i=0; i<Height; i++)
        for( int j=0; j<Width; j++)
            cin>>mat[i][j];
}// end function
/*This function will find the length of each river and inserting in
rList  [10 marks]*/

int* listOfRiverLengths(int mat[][Width], int size ) {
  int *rList= nullptr;
  // nested loop finds river at each index
  for(int i=0 ; i<Height ; i++)
    for(int j=0; j<Width; j++)
      if(mat[i][j]==1) // start of river found

     rList = 1 mat[i][j]                                  ;

  return rList;
}// end function
/*This function will expand the rList and insert the length of river
(rlength) in the rList [20 marks]*/

int* insertingRiver(int * rList , int size , int rlength){
    if(rList!=nullptr){

       *rList = rlength ;                                 ;

       for(int i=0; i < rlength                     ;  i++)

           ____________________________________________;

       ____________________________________________;

       ____________________________________________;
}
else
     rList=new int[  rlength                       ];

     rList[size - 1]= ____________________________________;

     return rList
}// end function                                          ;
```

04

```
/ This function is used to find length of a river and return its
length.Parameter 'x' and 'y' are river starting coordinates[14 marks]*/
int lengthOfRiver(int mat[][Width], int x, int y){
    mat[x][y] = 0; // by updating it with zero, means we don't want to
// visit the river again.
    int rlength = 1; // the minimum length of river will be 1
    bool flag = false;
    while(!flag){ // loop finishes when all neighbors are zero
        flag=true;
        if(x!=0 && mat[x - 1][y]){
            rlength ++;
            mat[x - 1][y]=0;

            x = ______________________________;

            flag = false______________________;

        }else if(____________________________________________){
            rlength ++;

            __________________________________;

            ______________________________________;

            flag = false______________________;

        }else if(____________________________________________){
            rlength ++;

            __________________________________;

            ______________________________________;

            flag = false______________________;

        }else if(____________________________________________){
            rlength ++;

            __________________________________;

            ______________________________________;
```

04

```
            flag = ____false____________________;
        }
    } // end loop
    return rlength;
} // end function
```

Rough Work