National University of Computer and Emerging Sciences

FAST School of Computing **Spring-2023** **Islamabad Campus**

PART I

Question 1 [Total Marks =20 Marks]

Multiple choice questions. Fill the correct option on the sheet Provided to you do not mark any option here. Marking multiple options will result in zero marks.

Answer Table

1. a. ○ b. ○ c. ○ d. ○ e. ○	11. a. ○ b. ○ c. ○ d. ○ e. ○
2. a. ○ b. ○ c. ○ d. ○ e. ○	12. a. ○ b. ○ c. ○ d. ○ e. ○
3. a. ○ b. ○ c. ○ d. ○ e. ○	13. a. ○ b. ○ c. ○ d. ○ e. ○
4. a. ○ b. ○ c. ○ d. ○ e. ○	14. a. ○ b. ○ c. ○ d. ○ e. ○
5. a. ○ b. ○ c. ○ d. ○ e. ○	15. a. ○ b. ○ c. ○ d. ○ e. ○
6. a. ○ b. ○ c. ○ d. ○ e. ○	16. a. ○ b. ○ c. ○ d. ○ e. ○
7. a. ○ b. ○ c. ○ d. ○ e. ○	17. a. ○ b. ○ c. ○ d. ○ e. ○
8. a. ○ b. ○ c. ○ d. ○ e. ○	18. a. ○ b. ○ c. ○ d. ○ e. ○
9. a. ○ b. ○ c. ○ d. ○ e. ○	19. a. ○ b. ○ c. ○ d. ○ e. ○
10. a. ○ b. ○ c. ○ d. ○ e. ○	20. a. ○ b. ○ c. ○ d. ○ e. ○

1. **Which of the following are the characteristics of Object Oriented programming?**
   a) Top-down design
   b) Complex code
   c) Global data focused.
   d) Complex design
   e) None
2. **Which of the following are the characteristics of Procedural Programming?**
   a) Protected data
   b) Bottom up design.
   c) Top-down design
   d) All of the above
   e) None
3. **In C++ memory management is done by:**
   a) Designer
   b) Programmer
   c) Automatically by IDE
   d) Garbage collector
   e) None
4. **Which of the following is the correct organization of virtual memory?**
   a) text, data, bss, heap, stack
   b) data, text, bss, heap, stack
   c) text, data, bss, stack, heap
   d) data, text, bss, stack, heap
   e) None
5. **What goes to data segment of virtual memory?**
   a) Program text/code
   b) Function's local variable
   c) Un-initialized globals & static data
   d) initialized globals & static data.
   e) None
6. **What goes to bss segment of virtual memory?**
   a) Program text/code
   b) Function's local variable
   c) Un-initialized globals & static data
   d) initialized globals & static data.

e) None

7. Choose the valid statement.

a) Ptr + Ptr
b) 1+ Ptr
c) 1- Ptr
d) 1 == Ptr
e) None

8. **What is the output of the following code?**

```scss
int arr[2][3] = {{1, 2, 3}, {4, 5, 6}};
int *ptr = arr[0];
cout << *(ptr + 4);
```

a) 5
b) 6
c) 7
d) 8
e) None

9. **What is the difference between a shallow copy and a deep copy in C++?**
   a) There is no difference between a shallow copy and a deep copy in C++.
   b) A shallow copy creates a new object with the same values, while a deep copy copies the values of the data members.
   c) A shallow copy only copies the public members, while a deep copy copies all the members.
   d) A shallow copy copies the values of the data members, while a deep copy creates a new object with the same values.
   e) None

10. **What is the output of the following code?**

```scss
int arr[2][2] = {{1, 2}, {3, 4}};
int (*ptr)[2] = arr;
cout << *(*ptr + 1);
```

a) 4
b) 3
c) 2
d) 1
e) None

11. What is the output of the following C++ code when executed?

```cpp
#include <iostream>

int myFunc(int n) {
    if (n > 0) {
        return n * myFunc(n-1);
    }
    else {
        return 1;
    }
}

int main() {
    int x = myFunc(-2);
    std::cout << x << std::endl;
    return 0;
}
```

a) 0
b) 1
c) -1
d) The code will result in a segmentation fault.
e) None

12. Which of the following is a potential issue when using recursion in C++?

a) Recursive functions can lead to more readable and maintainable code.
b) Recursive functions are typically faster than their iterative counterparts.
c) Recursive functions can result in a stack overflow error if the depth of the recursion is too large.
d) Recursive functions can only be used in situations where the input size is known in advance.
e) None

13. What is the output of the following code?

```cpp
#include <iostream>

void myFunc(int n) {
    if (n <= 0) {
        return;
    }
    else {
        std::cout << n << " ";
        myFunc(n-2);
        std::cout << n << " ";
    }
}

int main() {
    myFunc(7);
    return 0;
}
```

a) 7 5 3 1
b) 7 5 3 1 3 5 7
c) 7 5 3 1 1 3 5 7
d) The code will result in a segmentation fault.
e) None

14. What is the purpose of the "this" pointer in C++?

a) To access the value of a data member of the current object
b) To access the size of the current object
c) To access the type of the current object
d) To access the address of a data member of the current object
e) None

15. What is the difference between a structure and a class in C++?

a) A structure cannot contain member functions, while a class can
b) A structure is a primitive data type, while a class is a user-defined data type.
c) A structure has private access by default, while a class has public access by default.
d) A structure has public access by default, while a class has private access by default.
e) None.

16. Which type of data members must be initialized using member initializer list in C++?

a) Consts and references
b) Non consts
c) Non consts and references
d) All of the above
e) None

17. Which of the following statement is true about C++ destructors?

a) They are called automatically when an object is created.
b) They are called automatically when an object goes out of scope.
c) They can be called explicitly by the user.
d) They have a return type.
e) None

18. What is a static member function in C++?

a) A function that can only be called by objects of a class.
b) A function that is called by a static member of a class.
c) A function that can be called without an object of the class.
d) A function that is used to create static objects of a class.
e) None

19. What is the difference between public, private, and protected access specifiers in C++?

a) Public members can only be accessed by member functions of the same class, private members can be accessed by any function, and protected members can be accessed by derived classes.
b) Public members can only be accessed by derived classes, private members can only be accessed by member functions of the same class, and protected members can be accessed by any function.
c) There is no difference between public, private, and protected access specifiers in C++.
d) Public members can be accessed by any function, private members can only be accessed by member functions of the same class, and protected members can only be accessed by derived classes.
e) None

20. What is the purpose of the "new" operator in C++?

a) To allocate memory for an object of a class
b) To deallocate memory for an object of a class
c) To call the constructor of a class
d) To call the destructor of a class
e) None

Answer : 1- 10 [dcba seq] 11 [b] 12-15[cc,dd seq] [16-20] [abcd seq] [1 mark for each correct answer]

National University of Computer and Emerging Sciences

Question 2 [Expected time = 20 minutes]

1. **For the program given below identify which program segment goes to which virtual memory segment. Fill the table with program segments corresponds to each virtual memory segment. [12 marks for each correct answer] [7 minutes times]**

```
const int iSize=8;
int length;
int x = 123 ;
char *getx (int x)
{
  char *p;
  p = new char[iSize];

}
```

heap	stack	bss	text	data
p = new char[iSize]; [1 mark]	Char * p; Int x; [2marks]	Int length; 1 mark	Const int isize=8 1 mark	Int x= 123; 1 mark

2. What is the output of the following program? [5 marks]

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

int main(int argc, char** argv) {
    const int a = 5;
    const int *ptr;
    ptr = &a;
    *ptr = 10;
    cout << a << endl;
    return 0;
}
```

Output/Error if any

Compilation error [4 marks for correct answer else zer0]
because a is constant and ptr is changing its value so we get an error 2 marks for justification

3. What is the output of the following program? [5 marks]

```
#include <iostream>
using namespace std;
void function(char**);
int main()
{
    char *arr[] = { "ant", "bat", "cat", "dog", "egg", "fly" };
    function(arr);
```

```
        return 0;
}
void function(char **ptr)
{
        char *ptr1;
        ptr1 = (ptr += sizeof(int))[-2];
        cout <<  ptr1;
}
```

Output/Error if any

Cat [4 marks for correct answer else zer0]
as ptr will be incremented by the size of int and then -2 means to go back two position so first it will point at egg and after moving two positions back the output will be cat. [2 marks for justification]

4. What is the output of the following program? [5 marks]

```
#include <iostream>
using namespace std;
int main() {

   int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};
   cout << *(a[1] + 2) <<"  "<<  * (*(a + 1) + 2) ;
    return 0;
}
```

Output/Error if any

21 21 [4 marks if full correct 2 marks if half is correct]

The array a is a 2D array with dimensions 2x4.
The values inside the array are initialized as follows:
Row 0: 3, 6, 9, 12
Row 1: 15, 18, 21, 24
The expression *(a[1] + 2) is equivalent to accessing the element at index 2 in the second row of the array. So, it retrieves the value 21.
The expression *(*(a + 1) + 2) is another way of accessing the same element. It dereferences the pointer obtained by adding 1 to the base address of a (which points to the second row) and then adding 2 to that pointer

What is the output of the following program and also give reason for your answer? [5 marks]

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

int main(int argc, char** argv) {
        const char* str1 = "Best of Luck for the final Exam";
char str2[80]; //empty string
const char* src = str1;
char* dest = str2;
while( *src ) //until null character,
*dest++ = *src++;
*dest = '\0';
cout << *dest << endl;
        return 0;
}
```

Output/Error if any
Null/nothing [4 marks for correct answer else zer0]
As dest is char pointer so it is pointed to null character position at the end so no output will be shown. [2 marks for justification]

Reason:

Question 3 [Total Marks =25 Marks]

In this question you are required to code a data members of a data structure named connected list. Connected list is made of many nodes and each node in the connected list contains a data value (in your case it holds an integer value) and a reference or pointer to another node. In this way each element in the connected list is connected to the next one through a pointer or reference. The first node in the connected list is called the head node.

Each node has two parts: the data, which holds a number, and a pointer (represented by an arrow) to the next node in the list. The last node in the list points to null, indicating the end of the list.

Let's create an example connected list representing a sequence of numbers:

Node 1: [Data: 10, Next: Node 2]

Node 2: [Data: 45, Next: Node 3]

Node 3: [Data: 57, Next: NULL]

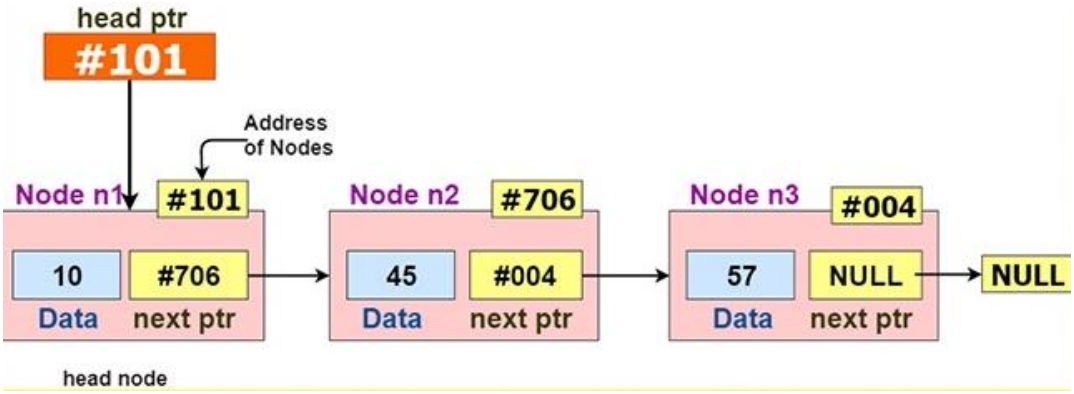
In this connected list, the first node contains the number 10, and its arrow points to Node 2. Node 2 holds the number 45 and points to Node 3, which holds the number 57. Its arrow points to null, indicating the end of the connected list. By following the arrows, we can traverse the connected list from the beginning to the end, accessing and processing each number along the way.

So, in this example, our connected list represents the sequence: 10 -> 45 -> 57.

A picture visualization of connected list is as follows.

National University of Computer and Emerging Sciences

FAST School of Computing **Spring-2023** **Islamabad Campus**



1. In this question you have to identify the attributes / data members of two blank classes given below. [Note: you are required to add only those data members that are explained in the above scenario, extra data members will result in the deduction of marks]. [5 marks]

class Node {

int data; [1.5 marks correct else zero]

Node *next; [1.5 marks correct else zero]

};

class Connected List {

Node * Head; [2 marks correct else zero]

};

Using the implementation of Node and Connected list class given above you are required to answer the following questions. Assume all dependencies are included.

2. Given a connected list with the following items: 3->2->4->3->8->2->NULL. What is the output of the following code. Also, what is purpose of the following magic function?

National University of Computer and Emerging Sciences


```
int magic_funcation(Node* l) {
   int result = 0;
   Node* current = l;

   while (current != nullptr) {
      result++;
      current = current->next;
   }

   return result;
}
 void main (){
       int result  = magic_funcation (head);
       cout << result
 }
```

Output

6 [5 marks correct else zero]

Purpose of the magic function

It is counting the number of nodes. [5 marks correct else zero]

3. Given a connected list with the following items: 3->2->4->3->8->2->NULL What will be the order of elements in the linked list after the execution of the following function? [10 marks]

```
Node* Function2 (Node* l, int value)
{
    if (l != null)
      if (l->value == value)
        l = l->next;
      else
        l->next = Function2(l->next, value);

    return l;
}

 void main (){


    head = Function2 (head, 8);

}
```

Answer:

3->2->4->3->2->NULL **[10 marks correct , 8 marks if 8 is removed and there is little mistake in the seq]**