

CS-1004: Object Oriented Programming

Serial No:

Final Exam**Total Time: 3 Hours****Total Marks: 150**Tuesday, 19th December, 2023

Course Instructor

Nimra Shahid

Signature of Invigilator

Student Name

Roll No.

Course Section

Student Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.**Instructions:**

1. Attempt on question paper. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work.
3. Verify that you have **twenty five (25)** different printed pages including this title page. There are **six (6)** questions.
4. Calculator sharing is strictly prohibited.
5. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.
6. Ensure that you do not have any electronic gadget (like mobile phone, smart watch, etc.) with you.

	Q-1	Q-2	Q-3	Q-4	Q-5	Q-6	Total
Marks Obtained							
Total Marks	23	42	10	20	20	35	150

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

Question 1 [6+ 5 + 6 + 3 + 3 = 23 Marks]

There are **NO ERRORS** in the following code. Show the output.

S no.	Question	Output
1.1	<pre>#include <iostream> using namespace std; int * func(int * n, int s) { int u = 0; for (int i = 0; i < s; i++) { if (i == 0 n[i] != n[i - 1]) u++; } int * r = new int[u]; int p = 0; for (int i = 0; i < s; i++) { if (i == 0 n[i] != n[i - 1]) r[p++] = n[i]; } return r; } int main() { int first[] = {3,4,4,5,7,7,8,12,12} ; int * second = func (first,9); for (int i = 0 ; i < 6 ; ++i) cout << second[i] << ", " ; return 0 ; }</pre>	3, 4, 5, 7, 8, 12, (1 mark each)

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

1.2	#include <iostream> using namespace std; int* mysterybox(int* p) { int i; for (i = 0; i < 2; i++) { int temp = (*p) + i; *(p + i) = (*p) + 3 - i; *(p + 3 - i) = temp; } return p; } int main() { int x[5] = { 1,2,3,4,5}; int* y = x; int i, *p; p = mysterybox(y); for (i = 0; i < 5; i++) { cout << (*p) + i << endl; } return 0; }	4 5 6 7 8 (1 mark each)
-----	--	--

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

1.3	<pre>#include <iostream> using namespace std; int shape (int count) { cout << count << endl; if (count < 5) shape(shape(++count)); return count; } int main() { cout << shape (3); return 0; }</pre>	<p style="color: red;">3 4 5 5 4</p>	<p>(1 mark each)</p>
1.4	<pre>#include <iostream> using namespace std; void flip(int* x, int* y, int*& z) { z = y; y = x; *x = 200; } int main() { int i = 10; int j = 20; int* p = &j; flip(&i, &j, p); cout << "i is =" << i << endl; cout << "j is =" << j << endl; cout << "p is =" << --(*p) << endl; }</pre>	<p style="color: red;">i is =200 j is =20 p is =19</p>	<p>(1 mark each)</p>

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

1.5	<pre>#include <iostream> using namespace std; void printNow(char *s) { if(*s) { cout<< *s; printNow(s+1); } } void printThen(char *s) { if(*s) { printThen(s+1); cout<<*s; } } int main() { char *str = "tit for tat"; printNow(str); cout << endl; printThen(str); cout<< endl; return 0; }</pre>	<p style="color: red;">tit for tat tat rof tit</p> <p>(1.5 mark each)</p>
-----	---	---

Question 2 [3 + 4 + 9 + 10 + 6 + 5 + 2 + 3 = 42 Marks]

There are **NO ERRORS** in the following code. Show the output.

S no.	Question	Output
2.1	<pre>#include <iostream> using namespace std; class Home{ Home *p; int NoOfRoom; public: Home () { NoOfRoom = 6; } Home (int NoOfRoom) : NoOfRoom(4) { cout << NoOfRoom << endl ;} Home (Home *x, const int NoOfRoom) : p(x) { this -> NoOfRoom = NoOfRoom; cout << NoOfRoom << endl; } Home& XYZ() { return *p; } void display(); }; void Home::display() { cout << p->NoOfRoom << endl; } int main () {</pre>	<p>3 8 6 (1 mark each)</p>

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

	<pre> Home r1, r2(3); Home *obj = &r2; Home r3 (obj, 8); r3.XYZ() = r1; r3.display(); return 0; } </pre>	
2.2	<pre> #include <iostream> using namespace std; class Base { protected: Base(){} cout<<"Base constructor"<<endl; } void show() { cout<<"in class A show"<<endl; } void access() { show(); } }; class Derived:Base { public: Derived(){} cout<<"Derived constructor"<<endl; } void access() { Base::access(); } }; class NDerived:public Derived { public: </pre>	<p style="color: red;"> Base constructor Derived constructor NDerived constructor in class A show </p> <p>(1 mark each)</p>

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

	<pre> NDerived(){ cout<<"NDerived constructor"<<endl; } }; int main() { NDerived obj; obj.access(); return 0; } </pre>	
2.3	<pre> #include <iostream> using namespace std; class Number { public: static int n; Number() {cout << n++ << endl; } Number(int i) { n=--i; cout<<n << endl; } static void somefunc(){ n=7; } Number(Number const &otherNum) { cout << otherNum.n++ << endl; } ~Number() {cout<<++n << " " ;} void fun(Number n) { cout<< n.n << endl; n.somefunc(); } int Number::n=6; int main() </pre>	6 4 4 5 6 8 9 10 11 (1 mark each)

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

	Number a, b(5), c(b); fun(a); return 0; }	
2.4	<pre>#include <iostream> using namespace std; class A { public: A(){ cout<<"A's constructor"<<endl;} A(int imaginary){ cout<<"A's parameter"<<endl;} ~A(){ cout<<"A's destructor"<<endl;} }; class B: public A { public: B(){ cout<<"B's constructor"<<endl;} B(int imaginary){ cout<<"B's parameter"<<endl;} B(A &b){ cout<<"B's copy constructor"<<endl;} ~B(){ cout<<"B's destructor"<<endl;} }; class C: public A { public: C(){ cout<<"C's constructor"<<endl;} C(int imaginary){ cout<<"C's parameter"<<endl;} C(B &a){ cout<<"C's copy constructor"<<endl;} ~C(){ cout<<"C's destructor"<<endl;} }; int main() { B* obj = new B(4); C c(*obj); delete obj; A obj2(4); return 0; }</pre>	<p style="color: red;">A's constructor B's parameter A's constructor C's copy constructor B's destructor A's destructor A's parameter A's destructor C's destructor A's destructor</p> <p>(1 mark each)</p>

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

<pre> 2.5 #include <iostream> using namespace std; class A { int x; public: A (int val=0): x(val) { cout<< "A " << x << endl; } A (const A &a) { x = a.x-1; cout<< "B " << x << endl; } void operator= (const A &a) { x = a.x-2; cout<< "D " << x << endl; } void setX(int x) { this->x = x; } int getX() const { return this->x; } }; void outA(A a) { cout << "OUT " << a.getX() - 2 << endl; } A InA(A &a) { cout << "Ref A " << a.getX() + 3 << endl; a.setX(a.getX() + 7); return a; } int main(){ A a(1), b = InA(a); a = b; outA(b); return 0; } </pre>	<p style="color: red;">A 1 Ref A 4 B 7 D 5 B 6 OUT 4</p>
---	--

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

	}	
2.6	<pre>#include <iostream> using namespace std; class A { public: A(int k){ cout<<k<<endl; }; class B: public A { public: B():A(3){ cout<<"B"<<endl; } }; class C: public A { public: C():A(5){ cout<<"C"<<endl; } }; class D: public B, public C { public: D(){ cout<<"D"<<endl; } }; int main() { D obj; return 0; }</pre>	<p style="color: red;">3 B 5 C D</p> <p>(1 mark each)</p>

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

2.7	<pre>#include <iostream> using namespace std; class A { int a; public: A(){ a=5; } static void display(int a) { cout<<a; } }; int main() { A obj; obj.display(15); return 0; }</pre>	15 (2 marks)
2.8	<pre>#include <iostream> using namespace std; class A { int a; public: A(int n):a(n) { } void display() const { cout<<a; } int getter(){ return a; } void setter(int k){ a=k;} }; class B {</pre>	13 (3 marks)

```
int a;
public:
B(int n):a(n)
{
}
void display() const
{
cout<<a;
}

int getter(){
return a;
};

class C
{
int a;
public:
C(int n):a(n)
{
}
void display() const
{
cout<<a;
}

int getter()
{
return a;
};

A adder(C &obj1, B &obj)
{
A temp(0);
temp.setter(obj1.getter()+obj.getter());
return temp;
}

int main()
{
A t1(2);
B t2(4);
C t3(9);
t1=adder(t3, t2);
t1.display();
return 0;
}
```

Question 3 [10 Marks]

Write a template function printArrayRange() that should receive an array, count of array elements, lower subscript and higher subscript. It should print array elements in the range. The function should Validate lowSubscript and highSubscript; if either is out of range or if highSubscript is less than or equal to lowSubscript, the function should display “range error” and terminate. Create a main function and call printArrayRange function for an integer, floating and char type array.

2 marks for correct template syntax

3 marks for correct function signature

1 mark for the validation of upper and lower case

2 marks for printing the array elements

2 marks for calling the function from main

Question 4 [20 Marks]

There are **NO ERRORS** in the following code. Show the output.

```
#include<iostream>
using namespace std;
class A {
protected:
    int a;
public:
    A() {
        a = 1;
    }
    virtual void func() {
        cout << "A's func" << endl;
        a++;
    }
    void myFunc() {
        a = 0;
    }
    int getA() { return a; }
    ~A() {
        cout << "A's destructor" << endl;
    }
};

class B :public A {
protected:
    int b;
public:
    B() {
        b = 10;
    }
    void func() {
        cout << "B's func" << endl;
        a++;
        b--;
    }
    void myFunc() {
        b = 100;
    }
    int getB() { return b; }
    virtual ~B() {
        cout << "B's destructor" << endl;
    }
};

class H {
```

```
public:  
    void func() {  
        cout << "H's func" << endl;  
    }  
    void myFunc() {  
        int a = 0;  
    }  
    ~H() {  
        cout << "H's destructor" << endl;  
    }  
};  
  
class C :public B,public H {  
public:  
    void func() {  
        a *= 3;  
        A::func();  
        cout << "C's func" << endl;  
    }  
    void myFunc() {  
        a += 5;  
        b -= 5;  
    }  
    ~C() {  
        cout << "C's destructor" << endl;  
    }  
};  
  
void main() {  
  
    A* a = new C;  
    a->func();  
    cout << a->getA() << endl;  
    delete a;  
    a = new B;  
    a->func();  
    a->myFunc();  
    cout << a->getA() << endl;  
    B* bptr = new C;  
    bptr->func();  
    cout << bptr->getB() << endl;  
    bptr->myFunc();  
    cout << bptr->getB() << endl;  
    delete a;  
    delete bptr;  
}
```

Output:

```
A's func    (1 mark)
C's func    (1 mark)
4    (1 mark)
A's destructor    (1 mark)
B's func    (1 mark)
0    (1 mark)
A's func    (1 mark)
C's func    (1 mark)
10    (1 mark)
100   (1 mark)
A's destructor    (2 marks)
C's destructor    (2 marks)
H's destructor    (2 marks)
B's destructor    (2 marks)
A's destructor    (2 marks)
```

Question 5 [20 Marks]

Imagine a publishing company that markets both books and audiocassette versions of its works. Create a class Publication that stores the title (string), and price (float) of a publication. From this class derive two classes: Book, which adds a page count (int), and Tape, which adds a playing time in minutes (float). Each of these three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display its data. Moreover, ensure these classes supports polymorphism. Now write a main function that creates an array of pointers to publication of size 5. In a loop, ask the user for data about a particular book or tape (complete information), and use new to create an object of type of Book or Tape to hold the data. Put the pointer to the object in the array. When the user has finished entering the data for all books and tapes, display the resulting data for all books and tapes

Entered, using a for loop and a single statement such as:

Pubarray[i] -> putdata();

To display the data from each object in the array.

3 marks for correct publication class

3 marks for correct book class

3 marks for correct tape class

2 marks for inheritance

5 marks for polymorphism

4 marks for main function

Question 6 [35 Marks]

Bank Management System is based on a concept of recording customer's account details. In BMS the user can perform all the tasks like creating an account, deposit amount, withdraw amount, check balance etc. In this question you are required to design a generic class **Bank** to hold the following information about a bank account:

- Balance
- Number of deposits
- Number of withdrawals
- Annual interest rate
- Monthly service charges

You need to select the data types of these attributes carefully.

The Bank class should have the following member functions:

- **Constructor:** Accepts arguments for the balance and annual interest rate.
- **Deposit:** A virtual function that will set the amount of the deposit and update the account balance accordingly. It should also set the variable holding the number of deposits.
- **Withdraw:** A virtual function that will set the amount of the withdrawals and update the account balance accordingly. It should also set the variable holding the number of withdrawals.
- **CalcInt:** A virtual function that updates the balance by calculating the monthly interest earned by the account, and adding this interest to the balance. This is performed by the following formulas:
 - Monthly Interest Rate = (Annual Interest Rate / 12)
 - Monthly Interest = Balance * Monthly Interest Rate
 - Balance = Balance + Monthly Interest
- **MonthlyReport:** A virtual function that subtracts the monthly service charges (Rs.850) from the balance, calls the calcInt function and finally sets the variables that hold the number of withdrawals, number of deposits, and monthly service charges to zero.
- **Display function.**

Next, design a **Savings account class**, derived from the generic Bankclass. The savings account class should have the following additional member:

- Status (to represent an active or inactive account)

If the balance of a savings account falls below \$25, it becomes inactive. (The status member could be a flag variable.) No more withdrawals may be made until the balance is raised above \$25, at which time the account becomes active again. The savings account class should have the following member functions:

- **Constructor:** Accepts arguments for the balance and annual interest rate.
- **Withdraw:** A function that checks to see if the account is inactive before a withdrawal is made. (No withdrawal will be allowed if the account is not active.) A withdrawal is then made by calling the base class version of the function.
- **Deposit:** A function that checks to see if the account is inactive before a deposit is made. If the account is inactive and the deposit brings the balance above \$25, the account becomes active again. The deposit is then made by calling the base class version of the function.
- **MonthlyReport:** Before the base class function is called, this function checks the number of withdrawals. If the number of withdrawals for the month is more than 4, a service charge of \$1

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

for each withdrawal above 4 is added to the base class variable that holds the monthly service charges. (Don't forget to check the account balance after the service charge is taken. If the balance falls below \$25, the account becomes inactive.)

- **Display function.**

Next, design a **Checking account class**, also derived from the generic Bank account class. It should have the following member functions:

- **Constructor:** Accepts arguments for the balance and annual interest rate.
- **Withdraw:** Before the base class function is called, this function will determine if a withdrawal (a check written) will cause the balance to go below \$0. If the balance goes below \$0, a service charge of \$15 will be taken from the account. (The withdrawal will not be made.) If there isn't enough in the account to pay the service charge, the balance will become negative and the customer will owe the negative amount to the bank.
- **MonthlyReport:** Before the base class function is called, this function adds the monthly fee of \$5 plus \$0.10 per withdrawal (check written) to the base class variable that holds the monthly service charges.
- **Display function.**

Write a main function that demonstrates these classes polymorphically by asking the user to enter the type of account, balance, and annual interest rate for an account. By using the concept of dynamic polymorphism perform the deposits, withdrawals and generate monthly report by calling the MonthlyReport() function of an account. Finally display all the statistics using the display function. **The display function must show the concept of pure polymorphism.**

For all these tasks make sure to follow OOP principles of Encapsulation, Inheritance, Abstraction, and Polymorphism.

Class Bank:

0.5 mark for parameterized constructor
1.5 mark for withdraw function
1.5 marks for deposit function
2.5 marks for calcInt()
3 marks for Monthly report function
3 marks for making display function pure virtual

Class Saving Account:

0.5 mark for parameterized constructor
1.5 mark for withdraw function
1.5 marks for deposit function
4.5 marks for Monthly report function
2 marks for making display function

Class Checking Account:

0.5 mark for parameterized constructor

1.5 mark for withdraw function

6 marks for Monthly report function

2 marks for making display function

3 marks for main function

Question 1 [4 + 4 + 3 + 4 = 15 Marks]

What will be the output of the given programs? In case of an error(s), circle that part in the code and mention the reason for that error(s) in one line.

S no.	Question	Output
1.1	<pre>#include <iostream> using namespace std; int a = 21; int b = 9; int *p = &a; int* func1(){ return &b; } int* func2 (int* p){ return p; } int& func2(){ return *p; } int& func3(){ return a; } int main() { int a = 8; int* p ; cout << *(func1()) << endl; p = func2(&::a); cout << *p+5 << endl; func2() = 1; cout << ::a+4 << endl; a= func3(); cout << a+2 << endl; return 0; }</pre>	9 26 5 3

1.2.	<pre>#include <iostream> using namespace std; int fun(int* p , int s) { p = new int [s]; for (int i=0; i<s ; i++) p[i] = (i+1)*2; return *p; } int main() { int *p; int x = fun(p,5); for (int i=4 ; i>0; i--) cout << x + i << endl; return 0; }</pre>	<p style="text-align: center;">6 5 4 3</p>
1.3.	<pre>#include <iostream> using namespace std; int* func(int *pi) { int a = 7; int **p2 = &pi; (*p2) = a+5; return *p2; } int main(){ int r= 10; int *p = &r; int* x = func(p); cout << (*p) + (*x); return 0; }</pre>	<p style="text-align: center;">24</p>
1.4.	<pre>#include <iostream> using namespace std; int main(){ bool x = 0; int y = 19; int *i= &y; void *ptr = &x; ptr = &y; cout << *ptr; const int* p = &y; (*p)++; cout << i; p = new int(0); cout << *p; return 0; }</pre>	<p>Error 1: Cannot dereference void pointer</p> <p>Error 2: P is a read only operator we cannot increment in its value</p>

--	--	--

Question 2 [4 + 3 + 6 + 4 = 17 Marks]

What will be the output of the given programs? In case of an error(s), circle that part in the code and mention the reason for that error(s) in one line.

S no.	Question	Output
2.1.	<pre>#include <iostream> using namespace std; int MyFunction(int s, int t) { if (t != 0){ s--; return (s*MyFunction(s, t-1)); } else{ return 1; } } int main(){ cout<<"Ouput "<<MyFunction(5,3); return 0; }</pre>	Output 24

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

int fun(int n, int* fp)
{
    int t, f;

    if (n <= 2) {
        *fp = 1;
        return 1;
    }
    t = fun(n - 1, fp);
    f = t + *fp;
    *fp = t;
    return f;
}

int main()
{
    int x = 15;
    cout << fun(5, &x) << endl;
    cout << x;
    return 0;
}
```

5
3

<pre> 2.3 #include <iostream> using namespace std; void print_asterisk(int asterisk) { if (asterisk == 0) return; cout << "^ "; print_asterisk(asterisk - 1); } void print_space(int space) { if (space == 0) return; cout << " " << " "; print_space(space - 1); } void pattern(int n, int num) { if (n == 0) return; print_asterisk(n); print_space(2 * (num - n) + 1); print_asterisk(n); cout << endl; pattern(n - 1, num); } int main() { int n = 5; pattern(n, n); return 0; } </pre>	
--	--

Question 2.4

Write a recursive function called powerofFive. Given an integer n, return true if n is a power of five. Otherwise, return false.

An integer n is a power of five, if there exist an integer x such that $n == 5^x$

Example1: Input n = 125

Output = true

Example1: Input n = 21

Output = false

```
bool powerofFive(int n)
{
    if (n==1)
        return true;
    else if ( n < 5)
        return false;
    else
        powerofFive(n/5);
}
```

```
int main()
{
    int n=125;

    cout<< powerofFive (n); //print true
    n=21;

    cout<< powerofFive (n); //print false

    return 0;
}
```

Question 3 [4+3+3=10 Marks]

What will be the output of the given programs? In case of an error(s), circle that part in the code and mention the reason for that error(s) in one line.

S no.	Question	Output
3.1.	<pre>#include <iostream> using namespace std; int main(){ int arr[] = {-6,8,5,0,8,3,5,7,12}; int* ptr; ptr = arr; arr[1] += 30; cout << "Pointer>" << *ptr << endl; *ptr -= 10; ptr++; cout << "Next" << ++(*ptr) << endl; ptr += 2; cout << "Last" << ptr[0] << endl; cout << "Final" << *(arr+4)*3 << endl; return 0; }</pre>	Pointer>-6 Next39 Last0 Final24
3.2	<pre>#include <iostream> using namespace std; int main() { const int x = 11; const int *const ptr = &x; int y = 15; const int *p = &x; int * const ptr2 = &y; ptr2 = p; cout<<*p<< " <<*ptr<< " <<*ptr2; return 0; }</pre>	Error: ptr2 cannot point to any other variable except y because it's a constant pointer.

3.3	#include <iostream> using namespace std; int main() { float data[] = { 10.2,20.0, 30.5, 40.5, 76.1}; double * a = new double; *a = *(data + 2); a = data; *a = (*a - *(a - 1)); cout << *(data+3); return 0; }	Error: pointer a is of double point. It cannot point to float type of array.
-----	---	--

Question 4 [8 Marks]

What will be the output of the given programs? In case of an error(s), circle that part in the code and mention the reason for that error(s) in one line.

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

struct someStruct
{
    int x = 1;
    int y = 10;
    int** z;
    someStruct* ptr;

}s1 = { 3,10,0,0 };

int main() {
    s1.z= new int* [s1.x+1];
    s1.z[0] = new int [4];
    s1.z[1] = new int [1];
    *(s1.z[0]) = -90;
    *(s1.z[0] +1) = -12;
    *(s1.z[1]) = 3;

    s1.ptr = new someStruct{ 2,86 };
    s1.ptr->z = new int* [s1.ptr->x];
    s1.ptr->z[0] = new int [4];
    s1.ptr->z[1] = new int [1];
    *(s1.ptr->z[0]) = 65;
    *(s1.ptr->z[0] + 1) = -2;
    *(s1.ptr->z[1]) = 87;

    s1.ptr->ptr = &s1;
    int i = 0, j=0;

    do{
        cout << *(*(s1.z+i)) << endl;
        cout << *(*(s1.ptr->z+j)) << endl;
        i++;
        j++;
    } while (i!=(s1.x-1));

    return 0;
}
```

-90

65

3

87

CS-1004: Object Oriented Programming

Serial No:

2nd Sessional Exam**Total Time: 1 Hour****Total Marks: 60**Monday, 6th November, 2023

Course Instructor

Nimra Shahid

Signature of Invigilator

Student Name

Roll No.

Course Section

Student 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 **thirteen (13)** different printed pages including this title page. There are total of **4** 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	Total
Marks Obtained					
Total Marks	23	6	25	6	60

Question 1 [2 + 5 + 2 + 2 + 5 + 2 + 2 + 3 = 23 Marks]

What will be the output of the given programs? In case of an error(s), circle that part in the code and mention the reason for that error(s) in one line. No need to correct the errors. Either write output or error, both will not be accepted.

S no.	Question	Output
1.1	<pre>#include <iostream> using namespace std; class XYZ{ int x,y; const int z; public: XYZ() : x(1), y(2), z(9) { x=0; y=0; z=0; } void display() { cout <<x << y << z; } }; int main() { XYZ xyz; xyz.display(); return 0 ; }</pre>	Error: We can only initialize the value of constant variable through initializer list therefore we cannot change the value of z through the body of constructor.

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

1.2.	#include <iostream> using namespace std; class Shape{ private: int side1, side2; public: Shape() : side1(8), side2(side1) { } Shape& operator() (int dx, int dy); void operator= (Shape s1) { cout << s1.side2 << endl; } void display () { cout << side1 << " " << side2 << endl; } }; Shape& Shape::operator()(int dx, int dy) { side1 = dy; side2 = dx; return *this; } int main() { Shape s; s(3,1)(4,5)(5,6); Shape s1(s); s1.display(); s = s1; s.display(); return 0; }	6 5 5 6 5
------	---	--

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

1.3.	<pre>#include <iostream> using namespace std; class Sample { int A; public: Sample() { A++; } static void fun() { A++; // Error: A is not static } }; int Sample::A = 0; int main() { Sample::fun(); return 0; }</pre>	<p>Error 1: we cannot use the non static (A) variable in static function</p> <p>Error 2: A is not a static variable therefore we cannot initialize it like this</p>
1.4.	<pre>#include <iostream> using namespace std; class Test { int A; public: Test() { A = 0; } Test(int a) { A = a; } Test operator+(Test A, Test B) { Test temp; temp.A = A.A + B.A; return temp; } void print() { cout << A << " "; } }</pre>	<p>Error 1: + operator is overloaded as a member function therefore there will be only parameter.</p>

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

	<pre> } }; int main() { Test T1(10); Test T2(20); Test T3; T3 = T1 + T2; T3.print(); return 0; } </pre>	
1.5	<pre> #include <iostream> using namespace std; class Integer { private: int *n; public: Integer() : n(new int(7)) { *n=5; } Integer(int nn):n(new int(4)) { *n=nn; cout << *n << endl; } Integer(Integer const& otherNum): n(otherNum.n) { cout << *n << endl; *n+=7; cout << *n << endl; } void display() { cout << *n << endl; } void increase() { *n *= 2; } }; int main(){ </pre>	3 3 10 20 20

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

	<pre> Integer a, b(3), c(b); b.increase(); c.display(); b.display(); } </pre>	
1.6	<pre> #include <iostream> using namespace std; int* fun() { static int a; a++; return &a; } static int a = 10; int main() { int* p; p = fun(); fun(); fun(); fun(); cout << "function called " << *p << " times"; return 0; } </pre>	Function called 4 times
1.7	<pre> #include <iostream> #include <stdlib.h> using namespace std; class Sample { public: static int A; Sample() { A++; } }; int Sample::A = 0; int main() { Sample S1, S2; } </pre>	3

National University of Computer and Emerging Sciences

FAST School of Computing

Fall-2023

Islamabad Campus

	<pre> new Sample(); cout << Sample::A; return 0; } </pre>	
1.8	<pre> #include <iostream> using namespace std; class Test { int VAL; public: Test(int v) { VAL = v; } Test* Sum(Test T1, Test T2) { VAL = T1.VAL + T2.VAL; return this; } void print() { cout << VAL << " "; } }; int main() { Test T1(10); Test T2(20); Test T3; T3 = T1.Sum(T3, T2); T1.print(); T2.print(); T3->print(); return 0; } </pre>	<p>Error 1: sum function is returning a pointer and T3 is not a pointer.</p> <p>Error 2: T3 is not a pointer therefore we cannot use arrow notation with it.</p>

Question 2 [6 Marks]

You are developing a dice game that involves rolling a six-sided die. Modify the dice class so that you can keep the track of how many times the game has been played. Zero marks will be awarded for creating the global variable.

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

class Dice{
int facevalue;

static int count;

public:
Dice ( int fv = 0)
{
facevalue = fv;

count++;

}

};

int Dice::count = 0 ;
```

Question 3 [25 Marks]

Consider the following code for class Dummy and main() method. No need to write the output.

Overload the following operators for Dummy class such that the compiler run the code without giving any error.

- 1) Unary minus operator: negate (change sign) of all member variables.
- 2) Binary divide operator: it will divide two dummy objects (divide x by x, divide y by y and divide z by z) Make sure that there is no zero integer value in the denominator when dividing the values. (if there is zero value in denominator then prompt the user that denominator is zero, hence you can't do the division operation)
- 3) Unary multiplication operator: multiply each member variable of the Dummy class by itself i.e. (x with x, y with y, z with z)
- 4) Stream insertion operator: Print all the values (x, y and z) of dummy object.
- 5) Stream extraction operator: Input all the values (x, y and z) of dummy object.

```
class Dummy
{
private:
    int x;
    int y;
    int z;
public:
    Dummy()
    {
        x=0;
        y=3;
        z=9;
    }
//operator overloading implementation
```

```
int main()
{
    Dummy d1, d2, d3;
    cin >> d1;
    cin >> d2;
    -d1;
    d3 = d1/d2;
    cout << d3;
    *d3;
    cout << d3;

    return 0 ;
}
```

Overloading as a member functions:

```
void operator - ( )
{
    x = -x;
    y = -y;
    z = -z;
}
```

```
void operator * ( )
{
    x = x*x;
    y = y*y;
    z = z*z;
}
```

Dummy& operator / (Dummy &d)

```
{
    if (d.x == 0 || d.y == 0 || d.z == 0)
    {
        cout << " We cannot do the division operation because denominator is zero";
        return *this;
    }

    else
    {
        Dummy temp;
        temp.x = x / d.x;
        temp.y = y / d.y;
        temp.z = z / d.z;
        return temp;
    }
}

friend ostream& operator << (ostream& , Dummy&);
friend istream& operator >> (istream& , Dummy&);

}; // class end
```

```
ostream& operator << (ostream& out , Dummy& d)
{
    out << d.x << d.y << d.z ;
    return out;
}
```

```
istream& operator << (istream& in , Dummy& d)
{
    in >> d.x;
    in >> d.y;
    in >> d.z;

    return in;
}
```


Question 4 [6 Marks]

What will be the output of the given programs? In case of an error(s), circle that part in the code and mention the reason for that error(s) in one line. No need to correct the errors. Either write output or error, both will not be accepted.

```
#include <iostream>
using namespace std;
struct A{
    int value;
    struct{
        int x = 4;
        int y = 2;} S;

    float y =9.2;

} A1;
```

0
4
0
10
4
0

```
class Test {
    int v;
    A a;

public:
    Test(int v)
    {
        this-> v = v;
    }

    Test (A a)
    {
        this -> a = a;
    }
}
```

```
void seta (A obj)
{
    A1.S.x = 50;
}

void print()
{
    cout << v << endl << a.S.x << endl << a.value << endl;
}
};
```

```
int main()
{
    Test T1(10);
    Test T2(A1);
    T2.seta(A1);

    A x = {6,2,3,4} ;
    Test *t = new Test(x);
    T2.print();
    T1.print();
    return 0;
}
```

-----Rough Work-----

CS-1004: Object Oriented Programming (CS)

Serial No:

Sessional Exam-II**Total Time: 1 Hour****Total Marks: 60**Monday, 10th April, 2023

Course Instructors

Amna Irum, Shams Farooq, Shehreyar Rashid,
Marium Hida

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 **ten (10)** different printed pages including this title page. There are a total of **2** questions. And write your instructor's complete office number in front of point 6 to secure bonus marks.
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	Total
Marks Obtained			
Total Marks	30	30	60

Question 01 [30 marks]

What would be the output produced by executing the following C++ codes? Identify errors, if any (either write output or error (syntax/runtime), both will not be accepted). All the code snippet contains #include<iostream> and using namespace std;

A [3 Marks]

```
class Box{
    int capacity;
    bool operator<(Box b){
        return this->capacity < b.capacity ? true : false;
    }
public:
    Box(){}
    Box(double capacity){ this->capacity = capacity; }
};
int main(){
    Box b1(10);
    Box b2(14);
    if(b1 < b2){
        cout<<"Box 2 has large capacity.";
    }
    else{
        cout<<"Box 1 has large capacity.";
    }
    return 0;
}
```

Output/Error:

Error: “Operator<” function is private.
Cant be accessed outside class.

B [5 Marks]

```
class Point{
    int x, y;
public:
    Point(int x=0, int y=0){
        this->x=x;    Point::y=y;
        (*this)();
    }
    void operator()(int y){
        cout<<" ("<<x<<","<<y<<") " <<endl;
    }
    Point& operator()(int y){
        this->y=y;
        return *this;
    }
    ~Point(){
        cout<<"Point is going";    (*this)();
    }
}p3;
int main() {
    Point *p=new Point(5,6);
    static Point p1(p3);
    p1(9)(8);
    delete p;
    Point p2(7);
    cout<<"-----"<<endl;
    return 0;
}
```

Output/Error:

(0,0
(5,6)
Point is going (5,6)
(7,0)

Point is going (7,0)
Point is going (0,8)
Point is going (0,0)

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

C [7 Marks]	
<pre> class Complex{ double r, i; public: Complex(double r=1.0,double i=1.0){ set(r,i); } void set(double r,double i){ Complex::r = r; this->i = i; } void print(){ if (i<0) cout << r << "" << i <<"i"<<endl; else cout << r << "+" << i <<"i"<<endl; } Complex operator+(Complex R){ Complex tmp; tmp.r = r + R.r; tmp.i = i + R.i; return tmp; } Complex operator++(){ Complex tmp=*this; r++; i++; return tmp; } Complex operator++(int){ ++(*this); return *this; } }; int main() { Complex A(3,4), B(5,-6); A.print(); B.print(); Complex C; C= A+B; C.print(); (++A).print(); C = ++A; C.print(); (A++) .print(); A.print(); return 0; } </pre>	<p>Output/Error:</p> <p>3+4i 5-6i 8-2i 3+4i 4+5i 6+7i 6+7i</p>

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

D [5 Marks]

<pre>class Mystery { public: static int n; Mystery() {cout << n++ << endl; } Mystery(int i) {n=i;cout << n << endl; } static void somefunc(){ n=5;} Mystery(Mystery const& otherNum){ n+=5; cout << n << endl; } ~Mystery(){cout << --n << " ";} }a; void fun(Mystery n){ cout << n.n << endl; n.somefunc(); } int Mystery::n=0; int main(){ Mystery b(9), c; fun(b); return 0; }</pre>	<p>Output/Error:</p> <p>0 9 9 15 15 4 3 2 1</p>
--	---

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

E [10 Marks]

```
class ZooCage
{
    int cageNumber;
    ZooCage* link;
public:
    ZooCage(int n):cageNumber(n),link(NULL){ }
    int getCageNumber() {return cageNumber;}
    ZooCage*& getLink() {return link;}
}*start=NULL;
void fun(ZooCage *& H, int num)
{
    if(H){
        fun(H->getLink(),num);
        return;
    }
    H = new ZooCage(num);
}
void fun(ZooCage * H)
{
    if(H){
        fun(H->getLink());
        cout<<H->getCageNumber()<<endl;
    }
}
int main(){
    fun(start,4);
    fun(start,2);
    ZooCage * temp=new ZooCage(5);
    temp->getLink()=start->getLink();
    start->getLink()=temp;
    fun(start,3);
    fun(start);
    return 0;
}
```

Output/Error:

3
2
5
4

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

Question 02 [30 marks]

Your goal is to write a program for creating a MovieStore. Your MovieStore should allow for storage of many movie names, there (**unique**) IDs and ratings. Your MovieStore should allow facility of performing following operations, (minimum) data members required are given (you can add more). You need to add the necessary functions as per sequence, following the sequence will give you 2 marks:

```
int main(){
    MovieStore s1, s2(10); // create two stores of size 5 and 10

    s1["Hobbit"] = 4.5;
    /* If previously not added, add a movie with average rating of 4.5
     * If already exist , overwrite rating by 4.5
     * If memory finished print "Memory finished overwriting rating of last movie"
     * And Update accordingly */

    s1["Lord of the Rings"] = 5;

    cout << s1[2] << endl;
    /* get rating for movie id 2 (Lord of the Rings => 5).
     * If id not found print "id not found"
     * And return -1 rating. */

    s2=s1; // copies elements from s1 to s2

    s1["Lion King"] = 8;
    cout << s2; // should display all the Movie store information
    return 0;
}
```

Expected Output:

5

MovieStore:

Id	Movie	Rating
1	Hobbit	4.5
2	Lord of the Rings	5

```
class MovieStore{
    string * names;
    int * ids;
    float * rating;
    int size;
    int currentSize;
    .....
```

```
#include <iostream>
```

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

```
#include <string>
using namespace std;
class MovieStore{
    string * names;
    int * ids;
    float * rating;
    int size;
    int currentSize;
public:

MovieStore(int s=5):size(s),currentSize(0){
    names=new string[size];
    ids=new int[size];
    rating=new float[size];
    for(int i=0;i<size;++i){
        names[i]="";
        ids[i]=i+1; // id has to be unique. This can also be done using other logics e.g. static variable
        rating[i]=0.0;
    }
}
float& operator[](string str){
for(int i=0;i<size;++i){
    if(names[i] == "") // if movie name not found, if will be true
    {
        names[i]=str;
        currentSize++; // increment of currentSize is must
        return rating[i];
    }
    else if (names[i]==str) // if movie name already exists
    {
        return rating[i];
    }
}
cout<<"Memory finished overwriting rating of last movie "<<endl;
return rating[size-1]; // memory capacity finished
}

float operator[](int id){
if(id>=ids[0] && id<=ids[currentSize-1]) // id has to be inside correct range
{
    // logic can be inform of loop (checking all ids)
    return rating[id-1];
}
else
{
    cout<<"id not found"<<endl;
    return -1.0;
}
}
```

```

void operator=(const MovieStore & c)
{
    if(size>0)
    {
        delete [] names;
        delete [] ids;
        delete [] rating;
    }
    size=c.size;
    currentSize=c.currentSize;
    names=new string[size];
    ids=new int[size];
    rating=new float[size];
    for(int i=0;i<size;++i){
        names[i]=c.names[i];
        ids[i]=c.ids[i];
        rating[i]=c.rating[i];
    }
}
~MovieStore() {
    delete[] names;
    delete[] rating;
    delete[] ids;
}

friend ostream& operator<<(ostream &,const MovieStore &);

// Note: without friend keyword students have to provide getters functions also
};

ostream& operator<<(ostream & out,const MovieStore & c)
{
    cout<<"MovieStore: "<<endl;
    cout<<"Id\t"=<<"Movie\t"=<<"Rating"=<<endl;
    for(int i=0;i<c.currentSize;++i){
        out<<c.ids[i]<<"\t"<<c.names[i]<<"\t"<<c.rating[i]<<endl;
    }
    return out;
}

int main(){
    MovieStore s1, s2(10); // create two stores of size 5 and 10
    s1["Hobbit"]= 4.5;
    /* if previously not added, add a movie with average rating of 4.5
     If already exist , overwrite rating by 4.5
     If memory finished print "Memory finished overwriting rating of last movie"
     And Update accordingly */

    s1["Lord of the Rings"] = 5;

    cout << s1[2]<<endl;
    /* get rating for movie id 2 (Lord of the Rings => 5).

```

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

If id not found print “id not found”

And return -1 rating.*/

```
s2=s1; // copies elements from s1 to s2
```

```
//cout << s1["Hobbit"]<<endl; // should print the movie hobbit ratings
```

```
s1["Lion King"] = 8;
```

```
cout << s2; // should display all the Movie store information
```

```
}
```

CS1004 Object Oriented Programming

Wednesday, May 31, 2023

Course Instructor

Dr. Bilal Khan, Dr. Imran Babar, Dr. Khalid Hussain, Rizwan ul Haq, Usman Ghous, Saud Arshad

Serial No:
Final Exam-Objective
Total Time: 40 Min
Total Marks: 35

Signature of Invigilator

Roll No

Section

Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Verify at the start of the exam that you have a total of two (2) questions printed on Nine (09) pages including this title page.
2. Attempt all questions on the question-book and in the given order.
3. The exam is closed books, closed notes. Please see that the area in your threshold is free of any material classified as 'useful in the paper' or else there may a charge of cheating.
4. Read the questions carefully for clarity of context and understanding of meaning and make assumptions wherever required, for neither the invigilator will address your queries, nor the teacher/examiner will come to the examination hall for any assistance.
5. Fit in all your answers in the provided space.
6. Use only your own stationery and calculator. **Calculator is not allowed.**
7. Use only permanent ink-pens. Only the questions attempted with permanent ink-pens will be considered. Any part of paper done in lead pencil cannot be claimed for checking/rechecking.
8. Subjective part can be given to the student upon submitting the Objective part at any time.

<Objective Part>

	Q-1	Q-2	Total
Total Marks	20	15	35
Marks Obtained			

Vetted By: _____ Vetter Signature: _____

University Answer Sheet Required: No Yes

Q1.

20

Fill the appropriate box with best answer for the MCQs provided.

Q. #	Answer			
1.	A	B	C	D
2.	A	B	C	D
3.	A	B	C	D
4.	A	B	C	D
5.	A	B	C	D
6.	A	B	C	D
7.	A	B	C	D
8.	A	B	C	D
9.	A	B	C	D
10.	A	B	C	D
11.	A	B	C	D
12.	A	B	C	D
13.	A	B	C	D
14.	A	B	C	D
15.	A	B	C	D
16.	A	B	C	D
17.	A	B	C	D
18.	A	B	C	D
19.	A	B	C	D
20.	A	B	C	D

1. The OOPs concept in C++, exposing only necessary information to users or clients is known as
 - A. Inheritance
 - B. Abstraction
 - C. Encapsulation
 - D. Polymorphism

2. What will be the output when an instance of class D is created:

```
class A {public: A() { cout << "A"; };};
class B : public A { public: B() { cout << "B"; } };
class C : public A { public: C() { cout << "C"; } };
class D : public B, public C { public: D() { cout << "D"; } };
```

- A. ABCD
- B. DCBA
- C. ABACD
- D. DCABA

3. What will be the output when an instance of class D is created:

```
class A {public: A() { cout << "A"; };};
class B : virtual public A { public: B() { cout << "B"; } };
class C : virtual public A { public: C() { cout << "C"; } };
class D : public B, public C { public: D() { cout << "D"; } };
```

- A. ABCD
- B. DCBA
- C. ABACD
- D. DCABA

4. What will be the output when an instance of class D is destroyed:

```
class A {public: ~A() { cout << "A"; };};
class B : public A { public: ~B() { cout << "B"; } };
class C : public A { public: ~C() { cout << "C"; } };
class D : public B, public C { public: ~D() { cout << "D"; } };
```

- A. ABCD
- B. DCBA
- C. ABACD
- D. DCABA

5. Constructor of a class cannot be _____

- A. virtual
- B. private
- C. friend
- D. None of the above is a correct answer

6. We cannot have overloaded

- A. Operators
- B. Destructor
- C. Constructor
- D. Function

7. A pointer to the base class can hold address of
 - A. Only base class instance
 - B. Only derived class instance
 - C. Base class instance as well as derived class instance
 - D. None of the above

8. How can a static member function be called in main function?
 - A. Using dot operator
 - B. Using arrow operator
 - C. Using dot or arrow operator
 - D. Using dot, arrow or using scope resolution operator with class name

9. Which of the following operator cannot be overloaded?
 - A. []
 - B. new
 - C. ::
 - D. +=

10. If same function is called from objects of several different classes and all of those can respond in a different way, what is this feature called?
 - A. Inheritance
 - B. Overloading
 - C. Polymorphism
 - D. Overriding

11. Which among the following best defines static data members?
 - A. Data which is common to all the classes
 - B. Data which is common to a specific method
 - C. Data which is allocated for each object separately
 - D. Data which is common to all the objects of a class

12. What is the memory address of this pointer
 - A. It does not have its own memory address as it is register variable
 - B. The address is the start of the object as it is part of object
 - C. Both A or B depending upon situation
 - D. None of the above

13. A constant member function can access
 - A. All members of the class
 - B. Only static members of the class
 - C. Only const members of the class
 - D. Only Static and const members of the class

14. Where does keyword ‘friend’ should be placed?

- A. function declaration
- B. function definition
- C. main function
- D. None of the mentioned

15. Which statement is true, if we execute the given code

```
int main() {  
    int *iPtr;  
    float b = 5;  
    iPtr = &b;  
    cout << *iPtr << endl;  
}
```

- A. It will print 5
- B. It will print address of b
- C. It will be a runtime error
- D. It will be a compiler error

16. What will be the result of Line 8

```
class Class1 { //Line 1  
    int x; //Line 2  
};  
class Class2 { //Line 3  
    int y = 15; //Line 4  
};  
int main() //Line 5  
{  
    Class1 c1; //Line 6  
    Class2 c2; //Line 7  
    c1 = c2; //Line 8  
}
```

- A. Will do nothing
- B. Will generate a syntax error
- C. Will generate a runtime error
- D. Will set the value of x in c1 to 15

17. A public member function foo() can be accessed by its pointer ptr in the following manner

- A. ptr->foo() only
- B. *ptr.foo() only
- C. *ptr.foo() and ptr->foo()
- D. ptr*.foo() and ptr->foo()

18. An array of the objects of the following class can be created as

```
class myClass {  
    int x;  
    int y;  
public:  
    myClass(int a, int b) {  
        x = a; y = b;  
    }  
};  
A. myClass *obj = new myClass[2] (5, 5);  
B. myClass *obj = new myClass[2] {5, 5};  
C. myClass *obj = new myClass[2] { (5, 5), (15,15)};  
D. myClass *obj = new myClass[2] { {5, 5}, { 15,15 }};
```

19. The relationship between Class1 and Class2 in the underneath given code is

```
class Class1 {  
    int a;  
public:  
    Class1() :a(0) {}  
};  
class Class2 {  
    int b;  
    Class1 *objClass1;  
public:  
    Class2(){  
        objClass1 = new Class1[5];  
    }  
    ~Class2() {  
        delete[] objClass1;  
    }  
};
```

- A. Composition
- B. Aggregation
- C. Inheritance
- D. Polymorphism

20. correct syntax for a function pointer for a function with prototype int foo(float a, int b); is

- A. int (*fPtr)(float, int)
- B. int *fPtr(float, int)
- C. int *fPtr(float a, int b)
- D. int (*fPtr)(float a, int b)

Q2.

$3*5 = 15$

Write down the output of following given program in the second column. **If the code contains some error or missing statement, write down the corrected line of code and the output after correction in the space provided.**

Sr.	Code	Output
a.	<pre> class A { public: A() { cout << 'A'; } ~A() { cout << "A1\n"; } }; class B:public A { public: B() { cout << 'B'; } ~B() { cout << "B1\n"; } }; class C:public A { public: C() { cout << 'C'; } ~C() { cout << "C1\n"; } }; int main() { A* ptrA[4]; for (int i = 0; i < 4; ++i) { if (i % 2) ptrA[i] = new B; else ptrA[i] = new C; cout << endl; } for (int i = 0; i < 4; ++i) { delete ptrA[i]; } return 0; } </pre>	<p>Output:</p> <p>AC AB AC AB A1 A1 A1 A1</p>

b.	<pre> class A { public: ~A() { cout << 'A' << endl; } }; class B :private A { public: ~B() { cout << 'B' << endl; } }; int main() { B b; A a; { B * Bptr; A * Aptr = &a; } return 0; } </pre>	Output: A B A
c.	<pre> class A { int a; public: A():a(10) {} int getVal() { return a; } ~A() { cout << 'A'; } }; class B :public A { int b; public: B():b(15) {} int getVal() { return b; } ~B() { cout << 'B'; } }; class C :public A { int c; public: C():c(20) {} ~C() { cout << 'C'; } virtual int getVal() { return c; } }; int main() { A* ptrA = new B; cout << ptrA->getVal() << endl;; ptrA = new C; cout << ptrA->getVal() << endl;; } </pre>	Output: 10 10

National University of Computer and Emerging Sciences

FAST School of Computer Science

Chiniot-Faisalabad Campus

d.	<pre> int foo(int *iPtr, int i) { if (i < 0) return 0; else if (i < 4) return iPtr[0] + *(iPtr - 2); else return *iPtr + foo(iPtr-1, i-1); } int main() { int arr[] = { 5,3,2,4,5,6 }; cout << foo(arr + 5, 5); } </pre>	Output: 18
e.	<pre> template<class T> void PrintSum(T a, T b) { cout << a + b; } template<class T, class U> void PrintSum(T a, U b) { cout << a + b; } int main() { int a = 2, b = 7; double c = 1.25, d = 2.25; PrintSum(a, b); cout << endl; PrintSum(c, d); cout << endl; PrintSum(a, c); cout << endl; PrintSum(c, a); cout << endl; PrintSum(10, 9); } </pre>	How many Instances of the function will be created: <u>2</u> Output: 9 3.5 3.25 3.25 19

CS1004 Object Oriented Programming

Wednesday, May 31, 2023

Course Instructor

**Dr. Bilal Khan, Dr. Imran Babar, Dr. Khalid
Hussain, Rizwan ul Haq, Usman Ghous, Saud
Arshad**

Serial No:

**Final Exam-
Subjective**

**Total Time: 2 hr 20
mins**

**Total Marks: 80
Marks**

Signature of Invigilator

Roll No

Section

Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Verify at the start of the exam that you have a total of five (5) questions printed on fifteen (15) pages including this title page.
2. Attempt all questions on the question-book and in the given order.
3. The exam is closed books, closed notes. Please see that the area in your threshold is free of any material classified as ‘useful in the paper’ or else there may a charge of cheating.
4. Read the questions carefully for clarity of context and understanding of meaning and make assumptions wherever required, for neither the invigilator will address your queries, nor the teacher/examiner will come to the examination hall for any assistance.
5. Fit in all your answers in the provided space. You may use extra space on the last page if required. If you do so, clearly mark question/part number on that page to avoid confusion.
6. Use only your own stationery and calculator. If you do not have your own calculator, use manual calculations.
7. Use only permanent ink-pens. Only the questions attempted with permanent ink-pens will be considered. Any part of paper done in lead pencil cannot be claimed for checking/rechecking.

	Q-1	Q-2	Q-3	Q-4	Q-5	Total
Total Marks	15	25	10	20	10	80
Marks Obtained						

Vetted By: _____ Vetter Signature: _____
University Answer Sheet Required: No Yes

Question 1:

CLO 2

Marks: 5

a. Rectify the given code. Only correct the erroneous lines. (5 marks)

```
class abc{
    private:
        int a;
    public:
        int abc (int b) { //constructor
            a = b;
            cout<<a<<"\n";
        }
        int sum (int x, int y){
            int mysum=x+y;
            cout<<"Sum: "<<mysum;
        };
int main() {
    abc s1(5), *s1;
    s1->abc(5);
    s1->sum(5,10);
    return 0;
}
```

abc (int b) { }	//Constructor should be without return type.	[Marks: 2]
abc s1(5), *s1;	//s1 and *s1 will result in error due to naming issue.	[Marks: 1]
s->abc(5);	// Constructor call using pointer is error	[Marks: 1]
s->sum(5,10);	// Function call using pointer	[Marks: 1]

b. What is the output of the given code snippet? (2 Marks)

```
class Count {
private:
int value;
public:
Count() : value(7){}
void operator ++() { value-=1;}
void display(){ cout<< "Count: "<<value<<endl;};
int main(){
Count objcount;
++objcount;
```

```
objcount.display();
return 0;
}
```

Count: 6

c. What is the output of the given code snippet? (3 Marks)

```
class base{
public:
~base() {
cout << "destructing base\n";
} };
class derived : public base {
public:
~derived() {
cout << "destructing derived\n";
} };
int main()
{
base *p = new derived;
delete p;
return 0;
}
```

Destructing base

d. Write the main function for the given code. Access all three show() member functions with one class pointer variable. (5 Marks)

```
class A {
public:
void show()
{
cout << "Parent class A ";
} };
class B : public A{
public:
void show()
{
cout << "Parent class B ";
} };
class C : public A{
public:
void show()
```

```
{
cout << "Parent class C ";
}};
```

```
class A {
public:
virtual void show() [Marks: 1]
{
cout << "Parent class A ";
};

int main()
{
A obj1;
B obj2;
C obj3;

A *Ptr; [Marks: 1]

Ptr = &obj1; [Marks: 1]
Ptr->show();

Ptr = &obj2; [Marks: 1]
Ptr->show();

Ptr = &obj3; [Marks: 1]
Ptr->show();

return 0;
}
```

Question 2:	Marks: 25
-------------	-----------

Implement a C++ program using object-oriented principles to count the occurrences of a specific element in an array. Design a template function that works for arrays of int, float, double, and char types. Your program should include the following:

1. Define a template function **CountOccurrences** that takes an array and the size of the array as arguments, along with a target element to count. **[CLO 4]** (5 Marks)

template<typename T, size_t N>

2. Implement the **CountOccurrences** function using a loop to iterate through the array elements, count the occurrences of the target element and return the count. **[CLO 1]** (5 Marks)

```
size_t CountOccurrences( const T (&arr)[N], const T& target )
{
    size_t count = 0;
    for (size_t i = 0; i < N; ++i) {
        if (arr[i] == target) {
            count++;
        }
    }
    return count;
}
```

3. Write a **main** function that demonstrates the usage of the **CountOccurrences** function by creating arrays of different types (**int**, **float**, **double**, and **char**) with varying sizes. Specify a target element for each array and call the **CountOccurrences** function to count the occurrences. [CLO 1] (5 Marks)

```
int main() {
    int intArr[] = {1, 2, 3, 4, 5, 2};

    int intTarget = 2;

    size_t intCount = CountOccurrences(intArr, intTarget);

}
```

4. Display the number of occurrences of the target element for each array type. [CLO 1] (5 Marks)

```

int main() {
    .
    .
    .
    size_t intCount = CountOccurrences(intArr, intTarget);

    cout << "Occurrences of " << intTarget << " in int array: " << intCount << endl;
}

```

5. Test your program with different arrays and target elements to verify its correctness and robustness. [CLO 1] (5 Marks)

```

int main() {
    int intArr[] = {1, 2, 3, 4, 5, 2};
    float floatArr[] = {1.1f, 2.2f, 3.3f, 2.2f, 4.4f};
    double doubleArr[] = {1.1, 2.2, 3.3, 4.4, 5.5, 2.2};
    char charArr[] = "Hello, World!";

    int intTarget = 2;
    float floatTarget = 2.2f;
    double doubleTarget = 2.2;
    char charTarget = 'o';

    size_t intCount = CountOccurrences(intArr, intTarget);
    size_t floatCount = CountOccurrences(floatArr, floatTarget);
    size_t doubleCount = CountOccurrences(doubleArr, doubleTarget);
    size_t charCount = CountOccurrences(charArr, charTarget);

    cout << "Occurrences of " << intTarget << " in int array: " << intCount << endl;
    cout << "Occurrences of " << floatTarget << " in float array: " << floatCount << endl;
    cout << "Occurrences of " << doubleTarget << " in double array: " << doubleCount << endl;
    cout << "Occurrences of " << charTarget << " in char array: " << charCount << endl;

    return 0;
}

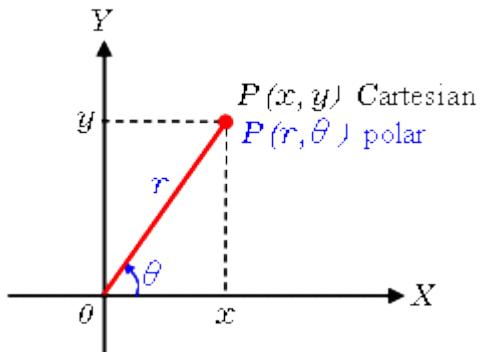
```

Question 3

CLO 3

Marks: 10

There are two ways to locate a point on the plane, Polar coordinates system and Cartesian coordinate system as shown in Figure below.



A point P in Polar system is defined as radius (r) and angle (theta), whereas the same point in Cartesian system can be represented by two numbers x and y . The following formula can be used to convert a polar point $P(\text{radius}, \text{angle})$ to the Cartesian Coordinate $C(x,y)$.

$$\begin{aligned}x &= \text{radius. cos(angle)}; \\y &= \text{radius. sin(angle)};\end{aligned}$$

You are required to create two classes, Polar and Cartesian. Both of the classes have private data members according to the explanation given above and there ***Does Not*** exist getter functions in both classes.

[3 Points for the Cartesian Class + 7 Points for the Polar Class]

Furthermore, your program shall provide the necessary functionality to successfully execute the following instructions given in main() routine.

```
int main () {
Polar p(10.5, 60)
Cartesian c = p;
}
```

There may exist two ways for conversion between the two coordinate systems, hence you are required to provide both solutions.

I asked for both ways to solve the conversion.

1. Conversion constructor to be implemented in Cartesian class

2. Using cast operator to be implemented in Polar class

```
class Cartesian {
```

```
private: //1 point
```

```
double x; double y;
```

```
public:
```

```
Cartesian(double _x=0, double _y=0) { x = _x; y = _y; }
```

```
//conversion constructor
```

Cartesian(Polar p) { //IMPORTANT: 2 Points. Conversion constructor shouldn't use any getter function for accessing polar's private members.

```
x = p.radius*cos(p.angle); y = p.radius*sin(p.angle); }
```

```
class Polar {
```

```
private: //1 Point
```

```
double radius; double angle;
```

```
public:
```

```
Polar (double r=0, double a=0) { //2 points
```

```
radius = r; angle = a;
```

```
}
```

```
friend class Cartesian; //2 points. Important for solution#1
```

```
operator Cartesian() { //2 Points, Solution using cast operator
```

```
double x = radius*cos(angle); double y = radius*sin(angle);
```

```
return Cartesian(x,y);
```

```
}
```

```
};
```

```
int main() {
```

```
Polar P(13, 60);
```

```
Cartesian c = P;
```

```
}
```

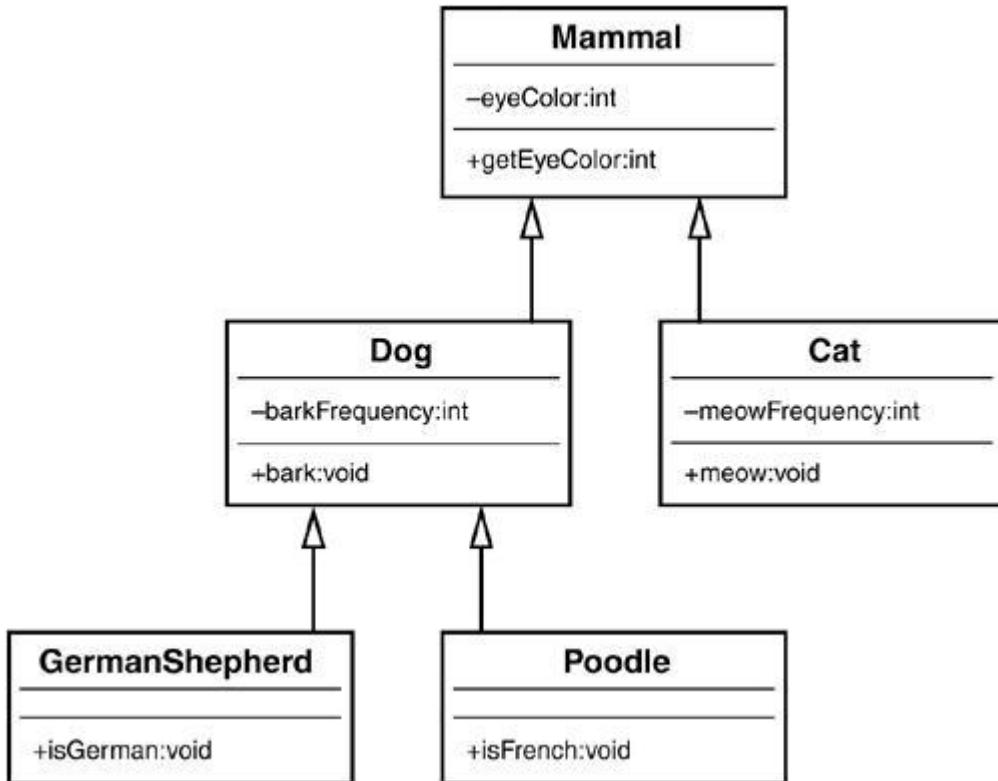
Question 4

Marks: 20

Part A: CLO 2

Implement the below UML diagram using the concept of polymorphism. The Mammal class is an abstract class, and it has getEyeColor method and destructor, both are pure virtual. In all derived classes there is default argument constructor and destructor. Call the base class methods in child classes and do overriding where necessary. [10 Marks]

In driver program, create an array of pointers of mammal class pointing to objects of derived classes. Use loop to call methods of derived classes with base class pointer. [4 Marks]



Driver Function (4 marks)

```
#include<iostream>
using namespace std;
// ***** CODE *****
// Step 1
class Mammal
{
int eyeColor;
public:
virtual int getEyeColor() = 0;
virtual ~Mammal() = 0;
// constructor
```

```
Mammal(){}
};

Mammal::~Mammal()
{
//cout << "Here is Pure Virtual Destructor of Mammal Class\n";
}

// Step 2
class Cat : public Mammal
{
int meowFrequency;
public:

// Default Argument Constructor
Cat(int meowFrequency = 6)
{
this->meowFrequency = meowFrequency;
}
void meow()
{
cout << "Bark Frequency : " << meowFrequency << '\n';
}

// function overriding
int getEyeColor() override
{
```

```
return 6;
}
//Destructor
~Cat(){}
};

// Step 3
class Dog : public Mammal
{
protected:
int barkFrequency;
public:
// Default Argument Constructor
Dog(int barkFrequency = 7)
{
this->barkFrequency = barkFrequency;
}
void Bark()
{
cout << "Bark Frequency : " << barkFrequency << '\n';
}
// Additional Function to Support Operator Overloading
int getBarkFrequency()
{
return barkFrequency;
}
//Destructor
~Dog() {}
};

// Step 4
class GermanShephard : public Dog
{
public:
void isGerman()
{
cout << "Yes! It's German\n";
}
// overriding
int getEyeColor() override
{
return 7;
}
//Destructor
~GermanShephard() {}
};
```

```

// Step 5
class Poodle : public Dog
{
public:
void isPoodle()
{
cout << "Yes! It's Poodle\n";
}
// overriding
int getEyeColor() override
{
return 8;
}
// Friend function to overload the stream insertion operator
friend ostream& operator<<(std::ostream& out, Poodle& p);
// Overloaded + operator as a member function
In driver program, create an array of pointers of mammal class pointing to objects of derived classes. Use loop
to call methods of derived classes with base class pointer. [4 Marks]
int main()
{
Mammal* mammal[3];
mammal[0] = new Cat();
mammal[1] = new GermanShephard();
mammal[2] = new Poodle();
for (int i = 0; i < 3; i++)
{
// Base Class Methods
cout << mammal[i]->getEyeColor() << endl;
}
// Poodle p1,p2;
// cout << p1;
// cout << p1 + p2 << endl;

```

```
// Deallocate Memory
for (int i = 0; i < 3; i++)
{
delete mammal[i];
}
return NULL;
}
```

Part B: CLO 4

6 Marks

1. Overload the stream insertion operators as friend function for the Poodle class to output the object of poodle class. [3 Marks]

// Definition of the friend function

```
ostream& operator<<(ostream& out,Poodle& p) {
out << "EyeColor: " << p.getEyeColor() << ", BarkFreq: " <<
p.getBarkFrequency() << "";
p.isPoodle();
out << endl;
return out;
}
```

2. Overload **+ operator** with return type integer and it should add barkFrequency of two Poodle objects. [3 Marks]

```
int operator+(const Poodle& other) const {
return barkFrequency + other.barkFrequency;
}
//Destructor
~Poodle() {}
};
```

Question 5:

CLO 2

Marks: 10

We want to store the information of different vehicles.

Create a class named Vehicle with two data member named mileage and price.

Create its two subclasses *Car with data members to store ownership cost, warranty (by years), seating capacity and fuel type (diesel or petrol). *Bike with data members to store the number of cylinders, number of gears, cooling type(air, liquid or oil), wheel type(alloys or spokes) and fuel tank size(in inches).

Make another two subclasses Audi and Ford of Car, each having a data member to store the model type.

Next, make two subclasses Honda and Yamaha, each having a data member to store the make-type.

Now, store and print the information of an Audi and a Ford car (i.e. model type, ownership cost, warranty, seating capacity, fuel type, mileage and price.)

Do the same for a Honda and a Yamaha bike.

Note: The solution below is very comprehensive and hence if portions of code in student's answer are true or partially right, marks can be awarded to the student.

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

class Vehicle{
private:
    float mileage;
    float price;
public:
    Vehicle(){}
    Vehicle(float mileage,float price){
        this->mileage=mileage;
        this->price=price;
    }
    float getMileage(){
        return this->mileage;
    }
    float getPrice(){
        return this->price;
    }
    virtual void display(){
        cout<<"Mileage: "<<mileage<<"\n";
        cout<<"Price: "<<price<<"\n";
    }
};

class Car:public Vehicle{
private:
    float ownershipCost;
    int warranty;
    int seatingCapacity;
    string fuelType;
public:
    Car(){}
    Car(float mileage,float price,float ownershipCost,int warranty,int seatingCapacity,string fuelType):Vehicle(mileage,price){
        this->ownershipCost=ownershipCost;
        this->warranty=warranty;
        this->seatingCapacity=seatingCapacity;
        this->fuelType=fuelType;
    }
    float getOwnershipCost(){
        return this->ownershipCost;
    }
    int getWarranty(){
        return this->warranty;
    }
}
```

```
int getSeatingCapacity(){
    return this->seatingCapacity;
}
string getFuelType(){
    return this->fuelType;
}
void display(){
    cout<<"Car\n";
    Vehicle::display();
    cout<<"Ownership Cost: "<<ownershipCost<<"\n";
    cout<<"Warranty: "<<warranty<<"\n";
    cout<<"Seating capacity: "<<seatingCapacity<<"\n";
    cout<<"Fuel type: "<<fuelType<<"\n";
}
};

class Bike:public Vehicle{
private:
    int numberCylinders;
    int numberGears;
    string coolingType;
public:
    Bike(){}
    Bike(float mileage,float price,int numberCylinders,int numberGears,string coolingType):Vehicle(mileage,price){
        this->numberCylinders=numberCylinders;
        this->numberGears=numberGears;
        this->coolingType=coolingType;
    }
    int getNumberCylinders(){
        return this->numberCylinders;
    }
    int getNumberGears(){
        return this->numberGears;
    }
    string getCoolingType(){
        return this->coolingType;
    }
    void display(){
        cout<<"Bike\n";
        Vehicle::display();
        cout<<"Number cylinders: "<<numberCylinders<<"\n";
        cout<<"Number gears: "<<numberGears<<"\n";
        cout<<"Cooling type: "<<coolingType<<"\n";
    }
};

class Audi:public Car{
private:
    string modelType;
public:
```

```

Audi(float    mileage,float    price,float    ownershipCost,int    warranty,int
seatingCapacity,string fuelType):
    Car(mileage,price,ownershipCost,warranty,seatingCapacity,fuelType){
        this->modelType="Audi";
    }
    void display(){
        Car::display();
        cout<<"Model type: "<<modelType<<"\n";
    }
};

class Ford:public Car{
private:
    string modelType;
public:
    Ford(float    mileage,float    price,float    ownershipCost,int    warranty,int
seatingCapacity,string fuelType):
        Car(mileage,price,ownershipCost,warranty,seatingCapacity,fuelType){
            this->modelType="Ford";
        }
    void display(){
        Car::display();
        cout<<"Model type: "<<modelType<<"\n";
    }
};

class honda:public Bike{
private:
    string makeType;
public:
    honda(float    mileage,float    price,int    numberCylinders,int    numberGears,string
coolingType):
        Bike(mileage,price,numberCylinders,numberGears,coolingType){
            this->makeType="Honda";
        }
    void display(){
        Bike::display();
        cout<<"The make-type: "<<makeType<<"\n";
    }
};

class yamaha:public Bike{
private:
    string makeType;
public:
    yamaha(float    mileage,float    price,int    numberCylinders,int    numberGears,string
coolingType):
        Bike(mileage,price,numberCylinders,numberGears,coolingType){
            this->makeType="Yamaha";
        }
    void display(){
}

```

```
Bike::display();
cout<<"The make-type: "<<makeType<<"\n";
}

};

int main()
{
    Audi carAudi(10000,36000,29000,10,3,"diesel");
    Audi carFord(20000,26000,19000,5,1,"petrol");
    honda bikeHonda(16000,12000,1,1,"liquid");
    yamaha bikeyamaha(56000,52000,6,6,"liquid");
    carAudi.display();
    cout<<"\n";
    carFord.display();
    cout<<"\n";
    bikeHonda.display();
    cout<<"\n";
    bikeyamaha.display();
    return 0;
}
```


CS-217: Object Oriented Programming

Serial No:

Final Exam

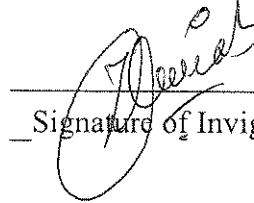
Total Time: 3 Hours

Total Marks: 140

Thursday, 8th July 2021

Course Instructors

Dr. Naveed Ahmad, Dr. Zainab Abaid, Mr. Rohail Gulbaz, Mr. Muhammad Usman


Signature of Invigilator

Aisha Irfan _____ 201-1851 _____ T _____ Aisha
Student Name Roll No Section Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Please read the paper carefully, set aside 10 minutes just to understand questions. Time every question and stick to it!
2. No additional sheet will be provided for rough work. There are two pages for rough work provided at the end of the paper.
3. After asked to commence the exam, please verify that you have thirty four (34) different printed pages including this title page. There are a total of five (5) questions.
4. Calculator sharing is strictly prohibited.
5. Smart device (of any sort) are not allowed.
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	Total
Marks Obtained	48.75	2	9	8.5	0	68.25
Total Marks	80	15	15	15	15	140

Question 1 [80 Marks]

Section A (20 marks) 13.5

- a) Which of the following codes are valid and invalid. Provide a short justification for your answer. (3 marks)

Code	Valid/Invalid - Justification
<pre>int main() { const int *p; const int a=2; p=&a; *p=7; return 0; }</pre>	INVALID Both integer and pointer are constant so value cannot be updated
<pre>int main() { const int a=2; const int *p=&a; int b=3; p=&b; return 0; }</pre> (3)	VALID P points to a const integer, so it cannot update value but can change the address it points to
<pre>int main() { int a[3]={1,2,3}; int * const p=a; cout<<*(p++); return 0; }</pre>	INVALID pointer is constant so it cannot be incremented

- b) Identify the errors (if any) in the following codes and fix them. (2+2=4 marks)

Code	Modified Code
<pre>int main() { int A[2][3]={{1,2,3},{4,5,6}}; int *p1,*p2; int B[3]={7,9,0}; p1=&B; //pointing to first element p2=A; //pointing to first element cout<<*p2; return 0; }</pre>	// Single pointer is pointed to 2-D array In array B we can do $p1 = B;$ In array A $\text{int } **p2;$ $p2 = \&A[0][0]$ (1)
<pre>int main() { void *vp;</pre>	

<pre> int a=6; float b=6.7; vp=&a; cout<<*vp; vp=&b; cout<<*vp; return 0; } </pre>	<p>No errors</p> <p>void pointer can point at any data type</p> <p>\times</p>
---	--

c) Write the output of the following programs. (2+2=4 marks)

Code	Output
<pre> int a=5; int b=6; int *p=&a; int * ABC() { return &b; } int * DEF(int *p) { return p; } int & DEF() { return *p; } int & GHI() { return a; } int main() { int a=4; int *p; cout<<*(ABC()); p=DEF(&::a); cout<<*p; } </pre>	<p>6411</p> <p>(1.5)</p>

```

DEF()=1;
cout<<::a;

a=GHI();
cout<<a;

return 0;
}

int main()
{
    char oop[]="OOP is an interesting course";
    char *pointer;
    pointer=oop;
    cout << "1 " << pointer[3]<<*pointer << endl;
    cout << "2 " << ++pointer << endl;
    cout << "3 " << ++(*oop) << endl;
    cout << "4 " << oop + 5 << endl;

    return 0;
}

```

(blank) OOP is an interesting course
 OOP is an interesting course

X

- d) Write the header of function ABC() for receiving 2D array in two different possible ways. Also, write the output for the below code. (1x3=3 marks)

Code:

```

void ABC//parameter to receive 2D array)
{
    for(int i=0; i<5; i++)
    {
        for(int j=0; j<5; j++){
            cout<<A[i][j];
        }
        cout<<endl;
    }

int main()
{
    int **A;
    A=new int*[5];
    for(int i=0; i<5; i++)
    {
        A[i]=new int[5];
        for(int j=0; j<5; j++){
            A[i][j]=i+j;
        }
    }
    ABC(A);
    return 0;
}

```

}

Header no.1: void ABC (int **A); ✓

Header no.2: void ABC (int *A[][5]); ✓

Output:

```

0 1 2 3 4
1 2 3 4 5
2 3 4 5 6
3 4 5 6 7
4 5 6 7 8

```

→ (3)

- e) Write a program to dynamically create, allocate, populate, print, and deallocate a 4D array using quadruple pointer. Avoid any dangling pointer in your program. (1x6=6 marks)

Write your code here:

```

#include <iostream>
using namespace std;
int main()
{
    int ****ptr; // creation
    ptr = new int ***[a];
    for (int i=0; i<a; i++)
    {
        ptr[i] = new int **[b];
        for (int j=0; j<b; j++)
        {
            ptr[i][j] = new int *[c];
            for (int k=0; k<c; k++)
            {
                ptr[i][j][k] = new int [d]; // memory allocation completed
                for (int m=0; m<d; m++)
                {
                    ptr[i][j][k][m] = k+m; // random population of array
                }
                cout << ptr[i][j][k][m]; // printing
            }
            cout << endl;
        }
    }
}

```

→ (5)

// quadruple pointers new int[a][b][c][d]

```

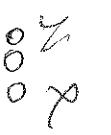
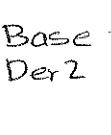
        for (int i=0; i<a; i++)
        {
            for (int j=0; j<b; j++)
            {
                for (int k=0; k<c; k++)
                {
                    ptr[i][j][k] = NULL;
                }
                delete [] ptr[i][j][k];
            }
            delete [] ptr[i];
        }
        delete [] ptr;
    }
}
    
```

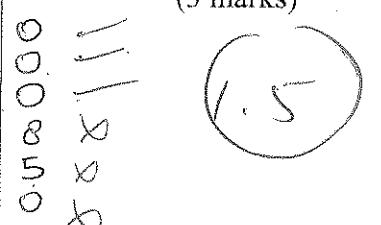
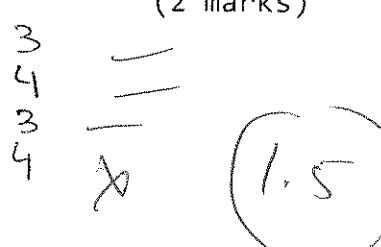
Section B (21 marks) M

- a) Write the output of the following programs (if any). If there is an error in the program, correct the code and then write the output.

	Code	Output
1.	<pre> class Seminar { int time; public: Seminar() //Function 1 { time = 30; cout << "Seminar starts now" << endl; } void lecture() //Function 2 { cout << "Lectures in the seminar on" << endl; } Seminar(int duration) //Function 3 { time = duration; cout << "Seminar starts now" << endl; } ~Seminar() //Function 4 { cout << "Thanks" << endl; } }; </pre>	<p style="text-align: center;">3</p> <p>(3 marks)</p> <p>Write statements in C++ that would execute Function 1, Function 3 and Function 4 of class Seminar.</p> <p>int main()</p> <p>Seminar S1;</p> <p>Seminar S2(15);</p> <p>delete S1;</p> <p>In Object Oriented Programming, which concept is illustrated by Function 1 and Function 3 together?</p> <p>overloading of constructors.</p>

<pre> 2. class student { int a; public: student (int i){ a = i; } void assign(int i){ a = i; } int return_value(){ return a; } }; int main() { student obj; obj.assign(5); cout<<obj.return_value(); } </pre>	<p style="text-align: right;">(2 marks)</p> <p>student () a = 3; //random</p> <p>Error : default constructor is not present</p> <p>Output : 5</p> <p style="text-align: right;">(2)</p>
<pre> 3. #include<iostream> using namespace std; class A { int i; public: A(int ii = 0) : i(ii) {} void show() { cout << i << endl; } }; class B { int x; public: B(int xx) : x(xx) {} operator A() const { return A(x); } }; void g(A a) { a.show(); } int main() { B b(10); g(b); g(20); return 0; } </pre>	<p style="text-align: right;">(2 marks)</p> <p>// function g only works for class A , it needs to be overloaded for class integers.</p> <p>After correction</p> <p>10 20</p> <p style="text-align: right;">(2)</p>

<pre> 4. #include <iostream> using namespace std; int i; class LFC { int x; int y; public: LFC(){ x = 0; cout << i << endl; } int getX(){ return x; } LFC getY(){ return *this; } ~LFC() { cout << i << endl; i=10; } }; int foo(LFC obj) { i=obj.getY().getX(); // cascaded function LFC ob; call return i; } int main() { LFC obj; cout << foo(obj) << endl; return 0; } </pre>	<p style="text-align: right;">(3 marks)</p> <p style="text-align: center;">    </p>
<pre> 5. class Base { public: Base () { cout << "Base" << endl; } Base (int i) { cout << "Base" << i << endl; } ~Base () { cout << "Destruct Base" << endl; } }; class Der : public Base { public: Der () { cout << "Der" << endl; } Der (int i) { cout << "Der" << i << endl; } ~Der () { cout << "Destruct Der" << endl; } }; int main () { Base a; Der d(2); return 0; } </pre>	<p style="text-align: right;">(3 marks)</p> <p style="text-align: center;"> Base  Der2  </p>

6. <pre>class C { public: C() : i(0) { cout << i << endl; } ~C() { cout << i << endl; } void iSet (int x) { i = x; } private: int i; }; int main() { C c1, c2; c1.iSet(5); { C c3; int i = 8; // invalid 'i' is private } corrects to return 0; c3.iSet(8); }</pre>	(3 marks) 
7. <pre>class A{ protected: int a; public: A (int x=1) { a = x; } int f() { a += 2; return a; } int g() { a += 1; return a; } int h() { f(); return a; } int j() { g(); return a; } }; class B : public A { private: int b; public: B (int y=5) { b = y; } int f() { b += 10; return b; } int g() { a += 3; return a; } }; int main() { A obj1; a = 1 B obj2; b = 5 cout << obj1.h() << endl; cout << obj1.g() << endl; cout << obj2.h() << endl; cout << obj2.g() << endl; }</pre>	(2 marks) 

- b) Write an operator overloading function for the following code. Assume getter and setter functions are available in each class, so you do not have to implement them. (3 marks)

```

class A
{
    int a, b;
public:
    A():a(5), b(10){
    }
};

class B
{
    int a, b;
public:
    B():a(6), b(10){
    }
};

class C
{
    int a, b;
public:
    C():a(7), b(10){
    }
};

void main()
{
    A test1;
    B test2;
    C test3;
    test1=test3+test2;
}

```

Write your code here:

A ~~B~~ :: operator+ (const & B right)

```

{
    A temp;
    temp.a = a + right.geta();
    temp.b = b + right.getb();
    return temp;
}

```

3

National University of Computer and Emerging Sciences 0241

FAST School of Computing

Fall 2019

Islamabad Campus

Section C (21 marks)

13

- a) What will be the output of the following code? If there are any errors, please correct them and then write the output. (3 Marks)

```
class Student {
    void print() {cout<<"Print details\n";}
    void display() {cout<<"Student\n";}
};

class GradStudent: public Student {
    void display() {cout<<"Grad Student\n";}
};

class Undergrad: protected Student {
    void Display() {cout<<"Undergrad Student\n";}
};

class TA: public GradStudent, public Undergrad { };

int main() {
    Undergrad stu;
    GradStudent gs;
    TA stuTA;
    cout<<"Line 1: "<<stu.display();
    cout<<"Line 2: "<<gs.display();
    cout<<"Line 3: "<<stuTA.Display();
    cout<<"Line 4: "<<stuTA.print(); will have to specify
    return 0;                                if
                                                stuTA.GradStudent :: print();
                                                or
                                                stuTA.Undergrad :: print()
}

```

Line 1 : Student
 Line 2 : Grad Student
 Line 3 : Undergrad Student
 Line 4 : Print details

(2)

- b) Do the following for the code below (6 marks)

```
class Book {
    int ID;
    char* name;
    Bookstore &bs;
public:
    Book(int id, char*n, bookstore) {
        this->ID = id; this->name = n; this->bs = bookstore;
    }
};

```

1. Is there any error in the above code? If yes, write the corrected code below. (2 marks)

*// Assuming that bookstore is a defined class
 an alias is created for Bookstore without an
 actual member bookstore*

(1)

2. What is the most likely relationship between the Book and Bookstore classes? Justify your answer. (2 marks)

Composition



3. Show how the parameterized constructor of the Book class will be called to create a Book object in main. (2 marks)

```
int n = 23;
char arr[20] = "Alchemist\0";
Bookstore B;
Book(n, arr, B); // call
```

✓ 2

- c) Do the following for the code below (6 marks)

Given the following classes:

```
class Base
{
    private:
        int first;
    Base() {cout<<"Default constructor of Base. ";}
protected:
    int second;
public:
    void display();
    Base(int s) {cout<<"Parameterised constructor of Base. "}
};

class Derived: public Base { };
```

1. In the class Derived, what would be the access level of: display() and first (2 marks)

first will be private; can only be accessed by getter/setter
display() will be publicly accessible.

✓ 2

2. Would the following statement execute in main, If not, explain how you would change the class definitions to allow this statement to work. (3 marks)

Derived d;

default constructor is private in Base class so

this will not work.

Page 12 of 34

we should put default constructor in public access

3. If instead we wrote class Derived: protected Base { }, would the following statement work in main? Justify your answer. (1 mark)

d.display(); //where d is an object of Derived that has been created before

No this will not work in main, since it is a protected function, it will not work outside class.

✓ (1)

d) Do the following for the code below (6 marks)

Given the following code:

```
class Document{
    //some members and functions
};

Document* createNewDocument() {
    Document *newDoc = new Document;
    return newDoc;
}

class Workbook {
private:
    int ID;
    static int num_books;
    Document *doc;
public:
    Workbook(int id) {this->ID = id;}
    void init() {
        doc = createNewDocument();
        num_books++;
    }
};

int main() {
    Workbook w(0);
    w.init();
    return 0;
};
```

1. Is there a compilation error in the above code? If yes, find and correct it. (2 marks)

Yes ✓

(2)
num_books has not been initialized outside class

CORRECTION

int Workbook :: num_books = 0;

2. What is the relationship between the Document and Workbook classes? Give a reason for your answer. (2 marks)

⇒ Association
⇒ pointer of Document is made ^y
⇒ The document does not depend on the Workbook.

3. Identify and correct a memory leak in this code. (2 marks)

Section D (19 marks) 8.25

- (a) Write a simple "Hello World" program that prints its own source code on the console. (3 marks)

```
fstream d ("file.txt", ios:: int out | app);  
while ( !feof() && !d.fail() )  
{  
    char ch;  
    d.get(ch)  
    cout << ch;  
}
```



3

- e) What will be the output of the following code? If there are any errors, please correct them and then write the output. (6 marks)

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

class Course{
public:
    virtual void Pro1(){cout<<"Prioritize ";}
    virtual void Pro2(){cout<<"your ";}
    virtual void Pro3(){cout<<"work ";}
    virtual void Con1(){cout<<"not fun ";}
};

class Programming:public Course{
public:
    virtual void Pro1(){cout<<"Programming ";}
    virtual void Pro2(){cout<<"is ";}
    void Pro3() const{cout<<"fun. ";}
    virtual void Con1(){cout<<"You have to do it! ";}
};

class AdvProgramming:public Programming{
public:
    virtual void Pro1(){cout<<"Com Prog 1 ";}
    virtual void Pro2(){cout<<"Com Prog 2 ";}
    void Pro3(){cout<<"Com Prog 3 ";}
    virtual void Con1(){cout<<"Com Prog 4 ";}
};

class DS:public AdvProgramming{
public:
    virtual void Pro1(){cout<<"has been ";}
    virtual void Pro2(){
        cout<<"the best. ";
        Programming::Con1();
    }
    void Pro3(){cout<<"DS 3 ";}
    virtual void Con1(){cout<<"DS 4 ";}
};

class Algo:public AdvProgramming{
public:
    virtual void Pro1(){cout<<"Algo 1 ";}
    virtual void Pro2(){cout<<"Algo 2 ";}
    void Pro3(){cout<<"Algo 3 ";}
    virtual void Con1(){cout<<"Algo 4 ";}
};

class BasicProg:public Programming{
private:
    virtual void Pro1(){cout<<"Learn basics ";}
    virtual void Pro2(){cout<<"was good ";}
    void Pro3(){cout<<"but ";}
    virtual void Con1(){cout<<"Bas Prog 4 ";}
};

class PF:public BasicProg{
public:
    virtual void Pro1(){cout<<"PF ";}
    void Pro3() const{cout<<"PF 3 ";}
    virtual void Con1(){cout<<"PF 4 ";}
};

class OOP:public BasicProg{
public:
    virtual void Pro1(){cout<<"OOP ";}
    virtual void Pro2(){cout<<"Reuse ";}
    void Pro3(){cout<<"Polymorphism ";}
    virtual void Con1(){cout<<"too many ";}
};

class Humanities:public Course{
public:
    virtual void Pro1(){cout<<"Important ";}
    virtual void Pro2(){cout<<"to ";}
    virtual void Pro3(){cout<<"learn ";}
    virtual void Con1(){cout<<"None ";}
};

class Isl:public Humanities{
};

class CommSkills:public Humanities{
};

int main()
{
    Course* cp; AdvProgramming* app; Course co;
    PF p; OOP o; Algo a; DS d; CommSkills c;
    Isl i;
    Course& cr = p; Course& cr1(co);
    Humanities h;
    cp = &p;
    cp->Pro1();          PF is fun
    cr = o;
    cr.Pro2();
    cr.Pro3();
    cp = &o;
    cp->Pro1();          OOP has been the best
    app = &d;
    app->Pro1();
    app->Pro2();
    cr1 = h;
    cr1.Pro1();           PF
    cr1.Pro2();
    cr1.Pro3();
    return 0;
}
```

Write the output!

PF is fun OOP has been the best Prioritize your work.

2.75

- (b) Do as mention in the following problems. If there are any errors, please correct them and then write the output

<pre> 1. #include<iostream> using namespace std; template<class T> // use <typename T> class myTemplate{ T* data; int size; public: myTemplate(){ data = NULL; // int, char, double // cannot be NULL size = -1; } myTemplate(T* obj, int s){ size = s; data = obj; // } void printData(); ~myTemplate(){} }; void myTemplate<T>::printData(){ for(int i=0; i < size; i++) cout << data[i] << endl; } int main(){ int* arr= new int[5]; for(int i=0; i < 5; i++) arr[i] = i; } </pre>	<p style="text-align: right;">(5 marks)</p> <p>Clearly mention all problems: logical or syntax</p> <p>template <class T></p> <p>data = NULL</p> <p>elements of array have to be copied one by one</p> <p>Deallocation is wrong</p> <p>Mention the fixes</p> <p>template <typename T></p> <p>data = 0</p> <p>data = new T [size]</p> <p>for (int i=0; i < size; i++)</p> <p>{</p> <p> data[i] = ob[i];</p> <p>}</p> <p>for (int i=0; i < 5; i++)</p> <p>{</p> <p> delete [] arr[i];</p> <p>}</p> <p>delete [] arr;</p> <p>Write the output</p>
---	---

	<pre> myTemplate<int> T(arr, 5); myTemplate<int> D(arr+2, 3); delete[] arr; // deallocation is wrong myTemplate<int> E(arr+5, 5); of 2d T.printData(); D.printData(); E.printData(); } </pre> <p style="text-align: center;">arr has been deleted</p>	<p>0 1 2 3 4 2 3 4 5 6</p> <p><i>garbage values</i></p> <p>2.5</p>
2.	<pre> #include<iostream> using namespace std; class A { public: virtual const char* fun1(int x) { return "A"; } virtual const char* fun2(int x) { return "A"; } }; class B : public A { public: virtual const char* fun1(short int x) { return "B"; } virtual const char* fun2(int x) const { return "B"; } }; int main() { B b; A& a(b); std::cout << a.fun1(1) << '\n'; std::cout << a.fun2(2) << '\n'; return 0; } </pre>	<p>Is there static polymorphism in this code? (yes/no) Write the statements where you think static polymorphism is?</p> <p>Write the output</p> <p>Is there any logical error? How can we fix it?</p>
3.	<pre> class A {public: void foo(){cout << "A";}}; class B:public virtual A{}; class C:public virtual A{}; class D:public B, public C{}; After the above code the hierarchy look something like: </pre>	<p>(2 marks)</p> <p>There are two paths now, how does it resolve the ambiguity problem?</p> <p>To call foo of class B</p> <p>D. obj; obj.B:: foo();</p> <p>To call foo of class C</p> <p>obj.C:: foo();</p>

Question 2 [15 Marks]

Part (a) Write a recursive program to generate the given pattern. Read a number 'n' and for 'n' equal to 3 the pattern will be as follows. (5 marks)

			-1		
		-4		-1	
	-9		-4		-1

Write your code here:

```
int main()
{
    int n;
    cout << "Enter your number for pattern";
    cin >> n;
    int arr [m][2*n-1];
```

Part (b) An ice-cream shop namely FROZEN is selling 5 flavors to customers in 5 cities. There are a total 10 outlets in each city. The customers are provided with an ID and top 10 regular customers get discount of 5% on family pack from each outlet. Suppose a customer can only visit one outlet. Regular customers become eligible for discount if they purchase ice-creams of minimum Rs. 2000 within a week and discount offer remains valid for the next week. Week begins from Sunday and ends on Saturday. Customers are sorted in descending order of their total purchasing worth. FROZEN calculates the total profit at the end of each month. Total production cost including tax is 70% of the selling price. Write a program to cover all details of FROZEN. (10 marks)

(2)

A table with prices is provided as:

Flavor	Size	Price
Mango	Mini	Rs.30/-
	Family Pack	Rs.120/-
Kulfa	Mini	Rs.35/-
	Family Pack	Rs.125/-
Strawberry	Mini	Rs.40/-
	Family Pack	Rs.130/-
Chocolate	Mini	Rs.40/-
	Family Pack	Rs.130/-
Vanilla	Mini	Rs.25/-
	Family Pack	Rs.110/-

Add any number of data members in the basic layout provided below or make more structures.

1. Make a function to calculate weekly purchase by a customer based on dates and days. (2 marks)
2. Make a function to print the names of top 10 customers in each outlet. (2 marks)
3. Provide a function to calculate and print the revenue generated by each outlet. (2 marks)
4. Provide a function to identify and print the age group more likely to receive the discounts. (2 marks)
5. Create a dynamic array of structures for 50 Frozen Outlets in the main function. (1 mark)
6. Pass structure to functions as a reference. (1 mark)

<pre> struct Address { string city; string location; }; struct Customer { string name; int custID; int purchase[5]; int age; }; struct Customer Record { int day; int month; int year; Purchase } </pre>	<pre> struct Flavor { int flavorID; int price[2]; }; struct FrozenOutlet { int OutletID; string *OwnerName; Address A; Customer TopCus[10]; Flavor item[5]; } int weeklyPurchase(FrozenOutlet); </pre>
--	--

```

int main()
{
    FrozenOutlet **fptr;
    fptr = new FrozenOutlet*[5]; // 5 objects
    for (int i=0; i<5; i++)
    {
        fptr[i] = new FrozenOutlet[10]; // DMA
    }

    Date CurrentDay;
    CurrentDay = 0;
    CurrentMonth = 7;
    CurrentYear = 2021;

    for (int i=0; i<5; i++)
    {
        for (int j=0; j<10; j++)
        {
            if (i==0)
                fptr[i][j].Address.city = "Islamabad";
            if (i==1)
                fptr[i][j].Address.city = "Lahore";
        }
    }

    PrintNames ( fptr );
}

```

①

```

int WeeklyPurchase (FrozenOutlet *f)
{
    void WeeklyPurchase (Customer T), Dated)
    int start = d.day - 7;
}

```

```

void printNames (FrozenOutlet *f)
{
    for (int i=0; i<5; i++)
    {
        cout << "CITY" << f[i][0].Address.city << endl;
        for (int j=0, j<10; j++)
        {
            cout << "LOCATION" << f[i][j].Address.location << endl;
            cout << "TOP 10 CUSTOMERS" << endl;
            for (int k=0, k<10; k++)
            {
                cout << k+1;
                cout << f.TopCustomer.name << endl;
            }
        }
    }
}

```

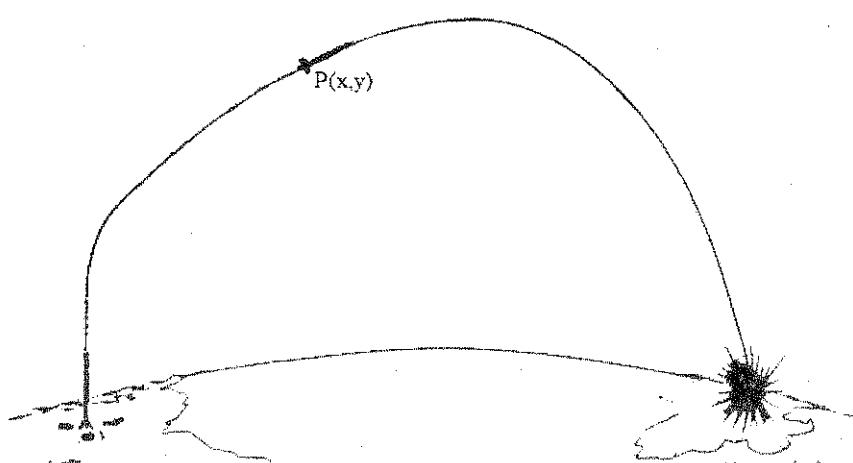
(1)

Question 3 [15 Marks]

The World has been involved in many full-fledged armed conflicts and cross-border ceasefire violations between various neighboring countries. You are also living in such a country which is surrounded by notorious neighbors doing violations, and various short ballistic missiles were fired in your territory. To increase your country's defense capability, government has decided to develop a system that intercepts the missiles fired by the enemies.

At any time, the missile can be fired from any random location with coordinates (x, y) . Your task is to intercept any missile fired towards your country with another missile (x, y) coordinates.

As the missile can be attack from any side, so their coordinates can also be negative.



Your program should implement the following:

1. Attack_Missile class that launch the enemy's missile. (3 marks)
2. Counter_Attack class that launches the intercepting missile. (3 marks)
3. Required Operator overloading functions which maps the coordinates of attacking missile and intercepting missile ($<=$, $>=$, $==$, $+$, $-$). (5 marks)
4. Show the difference between initial position and after intercepting position of attacked missile and counter attack missile and also shows the message on the screen after intercepting the missile. (4 marks)

Struct Location {

int x;

int y;

}

Class Attack {

public:

Location loc;

bool operator <= (const Attack & right);

bool operator >= (const Attack & right);

bool operator == (const Attack & right);

bool

Attack operator + (const Attack & right);

Attack operator - (const Attack & right);

}

```

bool Attack :: operator <= (const Attack & right)
{
    int check = 0;
    if (loc.x <= right.loc.x)
    {
        check++;
    }
    if (loc.y <= right.loc.y)
    {
        check++;
    }
    if (check == 2) ✓
        return 1;
    else
        return 0;
}

bool Attack :: operator >= (const Attack & right)
{
    int check = 0;
    if (loc.x >= right.loc.x)
    {
        check++;
    }
    if (loc.y >= right.loc.y) ✓
    {
        check++;
    }
    if (check == 2)
        return 1;
    else
        return 0;
}

```

```

body Attack :: operator == (const Attack & right)
{
    if (loc.x == right.loc.x && loc.y == right.loc.y)
        return 1;
    else
        return 0;
}

```

```

Attack Attack :: operator + (const Attack & right)
{
    Attack temp;
    temp.loc.x = loc.x + right.loc.x;
    temp.loc.y = loc.y + right.loc.y;
    return temp;
}

```

```

Attack Attack :: operator - (const Attack & right)
{
    Attack temp;
    temp.loc.x = loc.x - right.loc.x;
    temp.loc.y = loc.y - right.loc.y;
    return temp;
}

```

```

class Attack_Missile : public Attack {
}

```

```

class Counter_Attack : public Attack {
}

```

Question 4 [15 Marks] Q.S

You have to create a program to represent bill payment queues at a particular branch of a bank. There are four kinds of queues:

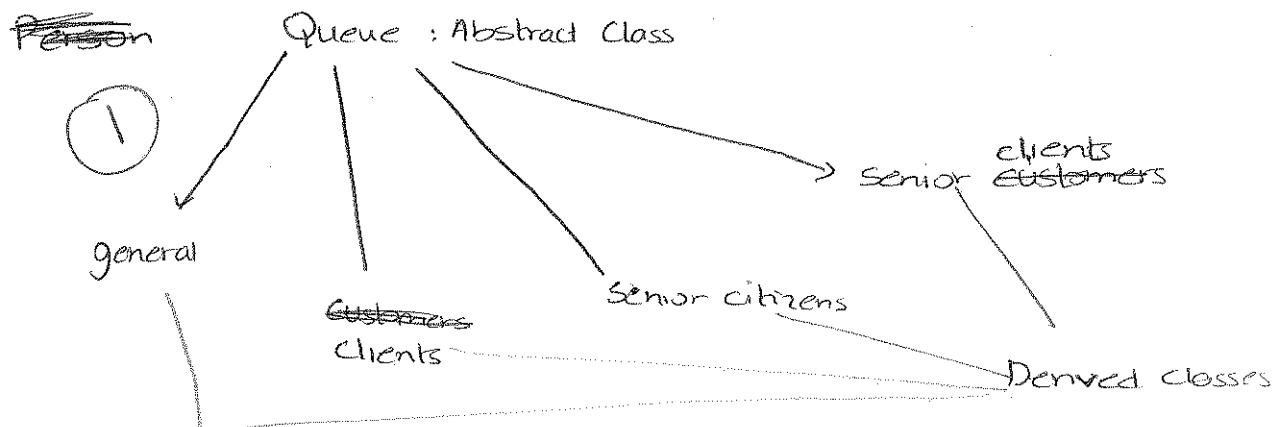
- one *general* queue, for all customers;
- one *client* queue, for customers who hold an account at that branch;
- one *seniors* queue, for senior citizens;
- and one *client-seniors* queue, for senior citizens who hold an account at that branch.

Each queue has a current length (current number of customers in queue) and a maximum length. The seniors queue has an age attribute that indicates after what age customers are eligible to enter this queue. The client queue has an attribute *branchCode* which indicates the branch code of the particular branch this program is instantiated for.

A receptionist receives incoming customers and assigns them to one of the queues. Each customer waits for their turn, pays their bill and leaves the queue. A manager can view the queue length for each queue and remove customers from anywhere in the queue (for example if a customer is being disruptive).

Implement an abstract **Queue** class to represent this scenario, with four concrete classes representing each type of queue. Use both single and multiple inheritance.

1. Draw class diagram showing all the classes and their functions that are needed to write a program to support the above scenario, and clearly indicate/mention type of relationship between classes. (2 marks)
2. Define the following functions in only the abstract class in a way that they can be accessed through objects of all the derived concrete classes.
 - (a) A *removeFromQueue()* function that deletes a customer (passed as parameter) from a queue, without changing the rest of the queue. This function should also update the queue length. (3 marks)
 - (b) A *getLength()* function that calculates the current number of customers in a queue. (2 marks)
 - (c) An *enqueue()* function that checks the status/type of a customer and inserts him into one of the three queues, updating its length appropriately. (2 marks)
3. Write a pure virtual out-of-line function *printQueue()* that should print the ID of each customer in a queue. Override it such that the each ID should be preceded by the name of the particular queue. For example, for a small seniors queue, the function should print:
seniors: ID 16
seniors: ID 25
seniors: ID 34
Define as many versions of *printQueue()* as needed to achieve this. (4 marks)
4. Now write a code segment that uses the **abstract class** to call the *printQueue()* function to print each of the three queues. (2 mark)



struct

~~class~~ Customer {

public:

int ID = -1;

bool account; // determine if customer is a client or not

string name;

int age;

Queue *q;

};

~~class~~ Queue {

Customer c[10]; // max people in 1 queue restricted to 10

int length;

void removeFromQueue();

{

int n;
cout << "Enter position of Customer in a Queue
who is to be removed." << endl;

cin >> n;

c[n].ID = -1;

}

void QueueLength()

{

length = 0;

for (int i=0 ; i<10 ; i++)

{

1.5

2

```

        if (c[i].ID != -1)
        {
            length++;
        }
    }

    cout << "Queue Length is : " << length << endl;
}

```

```

class general int getLen() { return length; }

virtual void printQueue() = 0

```

}

4

class general : public Queue {

```

void printQueue()
{
    for (int i=0; i<lo; i++)
    {
        if (ID != -1) (cout ((c[i].ID != -1))
            cout << "General" << c[i].ID << " " << c[i].name
            << endl;
    }
}

```

class Client : ~~general~~ public Queue {

```

void printQueue()
{
    for (int i=0; i<lo; i++)
    {
        if (c[i].ID != -1)
            cout << "Client" << c[i].ID << c[i].name << endl;
    }
}

```

}

class Senior : public Queue {

void printQueue()

client is changed to senior

class SeniorClient : public Queue {

client is changed to senior client

```
int main()
```

{

```
Queue * ptr1;
```

```
ptr1 = new general general;
```

```
Queue * ptr2;
```

```
ptr2 = new client;
```

```
Queue * ptr3;
```

```
ptr3 = new senior client;
```

```
Queue * ptr4;
```

```
ptr4 = new senior client;
```

~~cout << "Enter~~

do {

```
Customer cust;
```

```
cout << "Enter ID:";
```

```
cin >> cust.ID;
```

```
cout << "Enter 1 if you have an account else  
enter 0";
```

```
cin >> cust.account;
```

```
cout << "Enter age";
```

```
cin >> cust.age;
```

```
if (cust.age > 60 & & cust.account == 0)
```

{

```
ptr4->set[getlen()] = cust;
```

}

```
if (cust.age > 60 & & cust.account == 1)
```

{

```
ptr3->set[getlen()] = cust;
```

}

(On last page)

Question 5 [15 Marks]

You have to implement a Scene Generator for 2D games, where you have to randomly generate the gameAssets for a scene depending on the constraints entered by the user. Any scene consist of multiple assets. Figure below (LEFT) shows a typical scene, it has following assets: boy, fence, house, bushes, clouds, sidewalk, sky.



Figure: Coming to FAST for OOP Lecture at 8:30am (LEFT); After graduation (RIGHT)

In our game following gameAssets can exist: 1) a character; 2) hurdles, 3) background, and 3) textures. Character, Hurdles and Background are gameAssets and every asset has a texture. Backgroud is generated using multiple simpler assets, including: house, bushes, fence, cloud, sky. Hurdles are further classified into StaticHurdles of two types (Funcy, Tempy) and DynamicHurdles of two types (Pointy, Poly). Every scene generated needs to have a total of 7 gameAssets, where a character (*for example, boy in the scene*) and Background are mandatory. Your job as an OOP developer is to do the following:

1. Draw the class/object hierarchy (1 marks)
2. Implement classes, identified in the hierarchy. Top most class in the hierarchy (gameAssets) should be an abstract class. All classes should have constructors(), generateAsset() and destructors. DynamicHurdles and its sub classes should also have up(), down() functions for their movement. (3 marks)
3. Write a template class GameAssetArray which can hold gameAssets of any type. Other than the constructor(s) and destructor, you need to write two functions addGameAsset() and getGameAsset(). (5 marks)
4. Write a global function sceneGenerator() that return an object of GameAssetArray populated with 7 assets for a scene. You have to generate 7 gameAssets which must include a character and a background. (6 marks)

Following functions are dummy functions: generateAsset(), up(), and down(). You can simply write a cout statement in them printing an appropriate message.

ROUGH WORK

Continuation Q4

```
if (age < 60)
{
    if (cust cust.account = 1)
    {
        ptr2. C[getLen] = cust;
    }
    else
        ptr1. C[getLen] = cust;
}
```

CS-1004: Object Oriented Programming (CS)

Serial No:

Sessional Exam-I**Total Time: 1 Hour****Total Marks: 60**Monday, 27th February, 2023

Course Instructors

Amna Irum, Shams Farooq, Shehreyar Rashid,
Marium Hida

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 **nine (9)** different printed pages including this title page. There are a total of **3** questions. **And circle your instructor's name on page 1 to secure bonus marks.**
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	Total
Marks Obtained				
Total Marks	40	10	10	60

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

Question 01 [40 marks]

What would be the output produced by executing the following C++ codes? Identify errors, if any (either write output or error (syntax/runtime), both will not be accepted). All the code snippet contains #include<iostream> and using namespace std;

```
void mystery(int * ptr,int s)
{
    ptr=new int[s];
    for(int i=0, j=s;i<s;++i,j--)
        *(ptr + i)=j;
}
int main()
{
    int * ptr,s=5;
    mystery(ptr,s);
    for(int i=0;i<s;++i)
        cout<<ptr[i]<<" ";
    delete [] ptr; ptr=NULL;
    return 0;
}
```

Output/Error:

1. Memory leak
2. Segmentation fault

```
void function(char*** ptr)
{
    char* ptr1;
    ptr1 = (ptr += sizeof(int))[-2];
    cout<<ptr1<<endl;
}
int main()
{
    char* arr[] = { "ant", "bat", "cat", "dog", "egg", "fly" };
    function(arr);
    return 0;
}
```

Output/Error:

cat

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

Output/Error:

MID
s
Exam
OOP

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

<pre> struct structure { int x; structure *ptr; }; void print(structure * pointer) { while(pointer != NULL) { cout<<pointer->x<<" -> "; pointer=pointer->ptr; } cout<<"." <<endl; } int main() { structure three={ 10},two={ 30},one={ 20},*pointer=&one; three.ptr = &two; one.ptr = &three; print(pointer); structure four; four.x=15;four.ptr=pointer; pointer=&four; print(pointer); return 0; } </pre>	<p>Output/Error:</p> <p>20 -> 10 -> 30 -> . 15 -> 20 -> 10 -> 30 -> .</p>
<pre> void fun(int (*ptr)[3]){ cout<<ptr[1][2]<<" "; } int main() { int arr[9]={ 1,2,3,4,5,6,7,8,9 }; fun((int(*)[3])(&(arr))); fun((int(*)[3])(arr+1)); int arr2[3][4]={ 1,2,3,4,5,6,7,8,9,10,11,12 }; fun((int(*)[3])(arr2+1)); return 0; } </pre>	<p>Output/Error:</p> <p>6 7 10</p>

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

<pre>int main() { int array[2][5][2]={ 10,20,30,40,50,60,70, 80,90,100,18,21,3,4, 5,6,7,81,9,11}; int (*p)[5][2]; p=array; for(int i=0; i<2 ;i++) cout<<"\nthe vale is "<<*((int*)(p+1) + (1*2) + i); return 0; }</pre>	<p>Output/Error:</p> <p>the vale is 3 the vale is 4</p>
<pre>int main() { int number1 = 88, number2 = 22; int* pNumber1 = &number1; *pNumber1 = 99; cout << *pNumber1 << endl; cout << &number1 << endl; cout << pNumber1 << endl; cout << &pNumber1 << endl; pNumber1 = &number2; int& refNumber1 = number1; refNumber1 = 11; cout << refNumber1 << endl; cout << &number1 << endl; cout << &refNumber1 << endl; refNumber1 = number2; number2++; cout << refNumber1 << endl; cout << number1 << endl; cout << number2 << endl; return 0; }</pre>	<p>Output: (write “Address of **** Variable” where you identify the program will print address instead of writing some hypothetical address. **** will be replaced by the name of the variable whose address is assumed to be printed.)</p> <p>99 Address of number1 Address of number1 Address of pNumber1 11 Address of number1 Address of refNumber or number1 22 22 23</p>

```
int print_row(int ct, int num)
{
    if (num == 0)
        return ct;
    cout << ct << "\t";
    print_row(ct + 1, num - 1);
}
void pattern(int n, int count, int num)
{
    if (n == 0)
        return;
    count = print_row(count, num);
    cout << endl;
    pattern(n - 1, count, num + 1);
}
int main()
{
    int n = 5;
    pattern(n, 1, 1);
    return 0;
}
```

Output/Error:

```
1
2      3
4      5      6
7      8      9      10
11     12     13     14     15
```

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

Question 02 [10 marks]

Complete the following program as mentioned in the instructions and print the output as required.

```
int main() {  
    // use the following dimensions of an array  
    int X = 4;  
    int Y = 4;  
    int Z = 5;
```

Write code for dynamic allocation of a 3D array named **A**, using a triple pointer [2 marks]

```
int *** A= new int**[X];  
  
for(int i=0; i<X ; ++i )  
{  
    A[i]=new int*[Y];  
  
    for(int j=0; j<Y ; ++j )  
    {  
        A[i][j] = new int[Z];  
    }  
}
```

Assume that you have populated the array as following at each index using formula $i+j+k$; write the line within the given nested loop structure to assign those values to each index, using pointer notations.

[1 Marks]

```
for (int i = 0; i < X; ++i){  
    for (int j = 0; j < Y; ++j){  
        for (int k = 0; k < Z; ++k){
```

$$*(\ast(\ast(\mathbf{A} + \mathbf{i}) + \mathbf{j}) + \mathbf{k}) = i+j+k;$$

}

}

Use pointer arithmetic and print the output of the following statements. [5 marks]

Statement	Output
-----------	--------

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

cout << ***A << endl;	0
cout << *(*(*(A+1) + 2)) + 1 << endl;	4
cout << *(*(*(A+1)+2)+2) + 100 << endl;	105
cout << *(*(*(A + 2) + 2)) + 4 << endl;	8
cout << * (*(*(A + 2)))<< endl;	2

write your code to De-allocate the dynamic memory of the 3D array A [2 marks]

```
for(int i=0; i<X ; ++i )  
{  
    for(int j=0; j<Y ; ++j )  
    {  
        delete [] A[i][j];  
    }  
    delete [] A[i];  
}  
  
delete [] A;  
  
A=NULL;
```

```
return 0;
```

```
}
```

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

Question 03 [10 marks]

Complete the following C++ code that checks if a given char array has balanced parentheses or not.

Example01:

Input string: “00”
Output: true

Example02:

Input string: “(0)0”
Output: true

Example03:

Input string: “(00”
Output: false

Example04:

Input string: “)(”
Output: false

```
#include <iostream>
using namespace std;
bool isbalancedparentheses(char* s, int length, int open_count = 0) {
    if (open_count < 0) {
        return false;
    }
    if (length == 0) {
        return open_count == 0; // open_count == length
    }

    if (*s == '(') { // s[0] == '('
        return isbalancedparentheses(s + 1, length - 1, open_count + 1);
    }

    else if (*s == ')') { // s[0] == ')'
        return isbalancedparentheses(s + 1, length - 1, open_count - 1);
    }
    else {

        return isbalancedparentheses(s + 1, length - 1, open_count);
    }
}

int main() {
    char s[] = "()"; // Example string to check
    int length = sizeof(s) - 1; // Get the length of the string
    if (isbalancedparentheses(s, length)) {
        cout << "The string is balanced." << endl;
    }
    else {
        cout << "The string is not balanced." << endl;
    }
    return 0;
}
```

CS-1004 Object Oriented Programming (CS)

Date: May 20, 2024

Course Instructor(s):

Dr. Ali Zeeshan, Mr. Shehryar Rashid, Ms. Marium Hida

Final Exam

Total Time (Hrs): 3
Total Marks: 150
Total Questions: 5

Roll No

Course Section

Student Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Read the questions carefully, understand the question, and then attempt it on answer sheet.
2. Any answer provided on question paper will not be considered for marking. Therefore, avoid writing anything on question paper as final answer.
3. After asked to commence the exam, please verify that you have eight (8) different printed pages including this title page. There are a total of five (05) questions.
4. Calculator sharing is strictly prohibited.
5. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.
6. For additional 5 marks, please address all parts of a question in a single response. And mention the question and part no for each solution clearly on top of each answer. Write your roll number clearly on both answer sheet and question paper.
7. Use a permanent pen to clearly cross out any rough work.

Do not write below this line.

Attempt all the questions.

Q1. What would be the output produced by executing the following C++ code? Identify errors, if any (Either write output or error, both will not be acceptable). Please note that #include<iostream> and using namespace std; already included in code. [10+20+5+10=45 Marks]

a. class H {
public:
 H() { cout << " H-C "; }
 ~H() {
 cout << " H-D " << endl;
 }
};

class E {
public:
 E() { cout << " E-C "; }
 virtual ~E() { cout << " E-D "; }

```

};

class F : public E {
public:
    ~F() { cout << " F-D "; }

};

class A {
private:
    H h;
    E* ep;
public:
    A(E* t) : ep(t) { cout << " A-C "; }
    virtual ~A() {
        cout << " A-D ";
        delete ep;
    }
};

class B : public A, public F {
private:
    H h2;
public:
    B(E* t) : F(), A(t) { cout << " B-C "; }
};

int main() {
    B b(new E);
    cout << endl << " ----- " << endl;
    B b2(new F);
    cout << endl << " ----- " << endl;
    return 0;
}

```

- b. Please note that there are no syntax errors in this program

```

template<typename T, typename S>
class Matrice {
public:
    Matrice(int r, int c) :
        rows(r), columns(c), ptr(new T[r * c]) {}
    ~Matrice() {
        cout << *this;
        delete[] ptr;
    }

    void operator=(S* lmat) {
        for (int i = 0; i < rows * columns; ++i){
            ptr[i] = lmat[i];
        }
    }

    void operator ++(int dummy){
        for (int i = 0; i < rows * columns; i++)
            ptr[i]++;
    }

    T& operator()(int i, int j) {
        delete &(++this);
        return ptr[j + i * columns];
    }

    T& operator[](int i) { return ptr[i]; }
}

```

```

Matrice<T,S>& operator++() {
    Matrice<T,S> *M=new Matrice(columns,rows);
    *M=(this->ptr);
    for (int i = 0; i < rows * columns; i++)
        ptr[i]++;
    return *M;
}
int GetRows() { return rows; }

int GetCols() { return columns; }

private:
    int rows, columns; T* ptr;
};

template<typename T, typename S>
ostream& operator <<(ostream& cout, Matrice<T, S>& m) {
    cout << " Matrix = " << endl;
    for (int i = 0; i < m.GetRows() * m.GetCols(); ++i)
    {
        cout << m[i] << " ";
        if ((i + 1) % m.GetCols() == 0)
            cout << endl;
    }
    return cout;
}

int main() {
    int a[] = { 10, 30, 20, 40, 50, 55, 70, 80 };
    Matrice<int, int> v(4, 2);
    v = a;
    int z = ++v(2, 1);
    cout << " Z= " << z << endl;
    return 0;
}

```

```

c. struct node {
        int d;
        node* ptr;
};

class list {
    node* here;
public:
    list(node* p = NULL) : here(p) {}

    node* gethere() { return here; }

    void sethere(node* p) { here = p; }

    node* introduce(int x, node* here)
    {
        if (here == NULL)
        {
            here = new node;
            here->d = x;
            here->ptr = NULL;
            return here;
        }
    }
};

```

```

        }
        else
        {
            here->ptr = introduce(x, here->ptr);
        }
        return here;
    }

    void show(node* here)
    {
        if (here == NULL)
        {
            cout << endl;
            return;
        }
        else
        {
            cout << here->d << " ";
            show(here->ptr);
        }
    }

    node* wonder(node* here)
    {
        if (here == NULL || here->ptr == NULL)
            return here;
        node* here2 = wonder(here->ptr);
        here->ptr->ptr = here;
        here->ptr = NULL;
        return here2;
    }
};

int main()
{
    list d;
    d.sethere(d.introduce(73, d.gethere()));
    d.sethere(d.introduce(66, d.gethere()));
    d.sethere(d.introduce(91, d.gethere()));
    d.sethere(d.introduce(21, d.gethere()));
    d.sethere(d.introduce(13, d.gethere()));
    d.show(d.gethere());
    d.sethere(d.wonder(d.gethere()));
    d.show(d.gethere());
}
d. int x = 5;

class Carl {
    int x;

public:
    Carl(int x = 2) :x(x + 2) { cout << "Carl " << x << endl; }
    ~Carl() { cout << "~Carl " << this->x << endl; }
    int& getX() { return this->x; }
};

class Car2 {
    Carl* a2, a1;
    int x;
}

```

National University of Computer and Emerging Sciences

Islamabad Campus

```

    Car1 a2, a1 , m1 " 
public:           s
    Car2() :x(--::x), a1(--::x), a2(&a1) { cout << "Car2 " << x << endl; }
    Car2(int x) :a2(new Car1(3)), x(x), a1(x++) { cout << "Car2 " << x <<
endl; }           s   s ->
    ~Car2() { cout << "~Car2 " << Car2::x << endl; }
};

class Car3 {
    Car1& a;
    int x;
};

public:           q   s
    Car3(Car1& a) :x(a.getX()++), a(a) { cout << "Car3" << endl; }
    ~Car3() { cout << "~Car3" << a.getX() << " " << x << endl; }
};

class Car4 {
    Car3* c;      -?
    Car2* b;      -?
};

public:
    Car4(Car3* c) :c(c), b(new Car2) { cout << "Car4" << endl; }
    Car4(Car2* b, Car1* a) :b(b), c(new Car3(*a)) { cout << "Car4" << endl; }
    ~Car4() { cout << "~Car4" << endl; delete b; }
};

int main() {
    Car2 b(--::x);
    Car1 a;
    Car4 d1(&b, &a);
    cout << "-----" << endl;
    return 0;
}

```

Q2. For the given case study, generate a class diagram representing all classes of the system and what relationship exists between them. For each class, identify and include member variables and member functions. Use (+) to show public components, (-) for private component and (#) for protected components of the class. [20 Marks]

Blood donation management system is a web-based system that enables individuals who want to donate blood to help the needy. It also enables hospitals to record and store the data for people who want to communicate with them, and it also provides a centralized blood bank database. The system targets three types of users: the public who wants to donate blood, the recipients who need the donated blood, and the hospitals who work as an intermediary to manage the communication between the donors and recipients.

The main objective for developing the system is to educate the community on the benefits of blood donation, develop a web-based blood bank system to manage the records of donors and recipients, and encourage voluntary blood donation, easily accessing any information about blood type and the distribution of the blood in various hospitals, based on the hospital needs. This system provides various functionalities. Admin can observe and maintain the whole system namely from maintain donor details, update, and change database. Donors who can donate voluntarily have access to an individual account. Recipients view the system and search for a particular blood requirement and types in need. It facilitates donor registration and donor management by recording their physical and medical statistics. The

system has a facility to Search a blood donor via the internet by using social media like Twitter and Facebook. It also manages the list of donors who are eligible for donation on a particular date with contact number.

The inventory management in blood bank is responsible for storage and issuance of blood. It keeps records of blood requisition and issuance of blood. The salient feature is online request transferring of blood from one blood bank to another. By using this system, donors can view their donation records, including where and when they made donations, and the blood results for each, to learn of their donated blood quality and schedule their next donations. The system uses a First-In-First-Out stock management, where the blood stock that is checked-in to the system first will be the first one given to the hospital when requested. When the blood stock is expired, the system prompts for removing the stock from the inventory and updating the system.

The hospital can request blood via using the online blood bank system. This website will enable the public to make online reservations and includes online advertising for all the blood donation events. It will generate alerts to get all the details about the donation camps arranged by different groups. The system enables admin to generate a report to summarize all records including blood donation, blood requests and blood stock.

Q3. Analyze the following problem & write C++ recursive function.

[20 Marks]

Constraints:

You are not allowed to use loops or static/global variables in this problem. Helping function must also be recursive. 4 marks are reserved for function prototype and main function.

Flood fill:

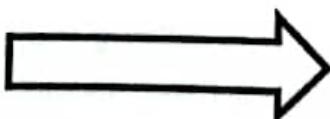
An image is represented by an integer grid image (row X col) where $\text{image}[i][j]$ represents the pixel value of the image.

You are also given six integers row (row size of image), col (col size of image), sr (row index), sc (col index), oldColor (previous color) and newColor (new color to be filled). You should perform a flood fill on the image starting from the pixel $\text{image}[sr][sc]$.

To perform a flood fill, consider the starting pixel, plus any pixels connected 4-directionally to the starting pixel of the same color as the starting pixel, plus any pixels connected 4-directionally to those pixels (also with the same color), and so on. Replace the color of all of the aforementioned pixels with color.

Example:

1	1	1
1	1	0
1	0	1



2	2	2
2	2	0
2	0	1

Input: image = [[1,1,1],[1,1,0],[1,0,1]], row = 3, col = 3, sr = 1, sc = 1, oldColor = 1, newColor = 2

Output: [[2,2,2],[2,2,0],[2,0,1]]

Explanation: From the center of the image with position [sr, sc] = [1, 1] (e.g. 1 means red pixel), all pixels connected by a path of the same color as the starting pixel are colored with the new color (e.g. 2 means blue pixels).

Note the bottom corner is not colored 2, because it is not 4-directionally connected to the starting pixel.

Q4: Your goal in this question is to design a mobile application management system in other words an "App Store". In the application store you can store information about different type of applications such as system and entertainment applications. For all the types of applications you will store application id (integer), programmer id (integer), and date of publishing. System applications include number of threads they can use and number of hardware devices they can access. Entertainment applications have predefined storage limit and already defined features. Apart from the applications information, "App Store" also stores user information, i.e. his name and email. Furthermore, it also records for each user what applications he is currently using.

[5+10=15 Marks]

- a. Now given the description, identify all the classes, their data members and their relationships (composition, aggregation, and inheritance) for the complete system.
- b. Implement the identified classes without their member functions. Main is not required.

Q5: Case Study: Academic Institution System.

[15+20+5=40 Marks]

You are tasked with designing a system for an academic institution that consists of different types of employees, including administrators, faculty members, and staff.

You have 6 classes in total, as listed below:

Employee	Administrator	Faculty
Staff	Professor	TeachingAssistant

Generally in an academic institution, all employees have a name, an ID number associated with them and different salaries, depending on their designation. However, certain job have additional information specific to their work that is required to maintain by the system.

An administrator administers a department, as they are responsible for day to day working of that specific department. Faculty is categorized on the basis of department as well, for example computer science faculty would come under the computer science department. While a Professor is a member of the faculty, they have a certain research area, in which they conduct their research over the years. Finally, in this system, the teaching assistant is given the primary task of teaching a course.

With the above description, you are required to design, develop and code the different classes, keeping in view each class' requirements. Please note that there cannot be any redundancy when it comes to attributes (multiple classes cannot have common attributes). Finally, while designing and coding your classes, ensure the following:

- Attributes of all the classes have *private* access modifiers. You are not allowed to create any new attributes, except for those mentioned above.
 - **No getter/setter functions are allowed.**
 - Every class must have a *display()* function that displays the *complete* information available.
 - *String class is not allowed.* Using strings in your code will result in severe penalization of marks.
 - Ensure that you *only provide methods that are necessary* for the proper functioning of the classes.
 - *Default constructor (i.e. a constructor with no parameters) is not allowed.* You must have a parameterized constructor.
 - Ensure that you handle *memory allocation/deallocation properly.*
 - You are encouraged to use C++'s *strcpy()* function, as described below:
 - `char* strcpy(char* dest, const char* src);`
- a. Draw a UML diagram representing the class hierarchy for the Academic Institution System. Illustrate the relationship between the aforementioned classes. Ensure that you have included the necessary attributes and methods in your UML diagram. Generate a class diagram representing all classes of the system and what relationship exists between them. For each class, identify and include member variables and member functions. Use (+) to show public components, (-) for private component and (#) for protected components of the class.
 - b. Write C++ code for the classes described in the UML diagram.
 - c. Demonstrate polymorphic behavior by creating objects of different employee types and calling their display functions in the *main()* function. Create a single array of objects, store the information as listed in the table below and display it using a for-loop.

Object Type	Name	ID	Salary	Department	AdditionalInformation
Professor	John Doe	P001	80000	CS	AI (research area)
TeachingAssistant	Jane Smith	TA001	30000	Mathematics	Calculus (course taught)
Administrator	Jack Black	AD001	60000	Administration	-

Bonus Question [5 Marks]

Write the correct answer on the first page of your answer sheet. Clearly mention "Bonus" on top of your answer.



Crack the code.

6	4	5
---	---	---

Nothing is correct

5	8	4
---	---	---

One number is correct,
but wrongly placed

7	2	8
---	---	---

Two numbers are
correct. Both wrongly
placed

3	8	6
---	---	---

Two numbers are
correct, but wrongly
placed

8	7	3
---	---	---

Two numbers are correct.
One is well placed, and the other
wrongly placed



CS-1004: Object Oriented Programming (CS)

Monday, 27th February, 2023

Course Instructors

Amna Irum, Shams Farooq, Shehreyar Rashid,

Mariam Hida

Serial No:

Sessional Exam-I

Total Time: 1 Hour

Total Marks: 60

Signature of Invigilator

Ancez Ahmed Malik 21-1167 CS-B

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 nine (9) different printed pages including this title page. There are a total of 3 questions. And circle your instructor's name on page 1 to secure bonus marks.
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	Total
Marks Obtained	22	10	08	40
Total Marks	40	10	10	60

40+8=48

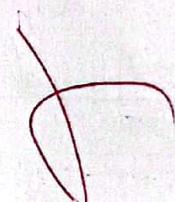
Question 01 [40 marks]

What would be the output produced by executing the following C++ codes? Identify errors, if any (either write output or error (syntax/runtime), both will not be accepted). All the code snippet contains #include<iostream> and using namespace std;

```
void mystery(int * ptr,int s)
{
    ptr=new int[s];
    for(int i=0, j=s;i<s;++i,j--)
        *(ptr + i)=j;
}
int main()
{
    int * ptr,s=5;
    mystery(ptr,s);
    for(int i=0;i<s;++i)
        cout<<ptr[i]<<" ";
    delete [] ptr; ptr=NULL;
    return 0;
}
```

Output/Error:

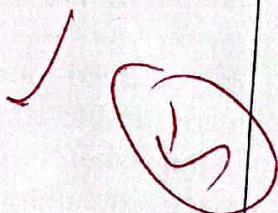
5 4 3 2 1



```
void function(char** ptr)
{
    char* ptr1;
    ptr1 = (ptr += sizeof(int))[-2];
    cout<<ptr1<<endl;
}
int main()
{
    char* arr[] = { "ant", "bat", "cat", "dog", "egg", "fly" };
    function(arr);
    return 0;
}
```

Output/Error:

cat

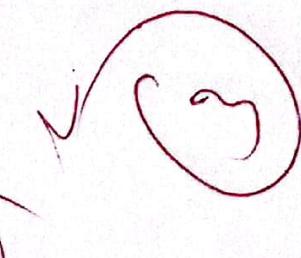


```
const char* c[] = { "OOP", "Exam", "Opsmid-1", "MID" };
char const ** cp[] = { c + 2, c + 3, c, c + 1 };
char const *** cpp = cp;
int main()
{
    cout << *cpp[1] << endl;
    cout << *(*(*(cpp + 2) + 2) + 3 ) << endl;
    cout << (*cpp)[-1] << endl;
    cout << *(cpp + 3)[-1] << endl;
    return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

Output/Error:

MID

Exam
Exam



```

struct structure {
int x;
structure *ptr;
};
void print(structure * pointer)
{
while(pointer != NULL)
{
cout<<pointer->x<<" -> ";
pointer=pointer->ptr;
}
cout<<". "<<endl;
}
int main()
{
structure three={10},two={30},one={20},*pointer=&one;
three.ptr =&two; one.ptr = &three;

print(pointer);
structure four;
four.x=15;four.ptr=pointer;
pointer=&four;
print(pointer);
return 0;
}

```

Output/Error:

~~20 -> 10 -> .
15 -> 20 -> 10 -> .~~

4

```

void fun(int (*ptr)[3]){
cout<<ptr[1][2]<< " ";
}
int main()
{
int arr[9]={1,2,3,4,5,6,7,8,9};
fun((int(*)[3])(&(arr)));

fun((int(*)[3])(arr+1));

int arr2[3][4]={1,2,3,4,5,6,7,8,9,10,11,12};
fun((int(*)[3])(arr2+1));

return 0;
}

```

Output/Error:

6 7 10

✓

5

```
int main()
{
    int array[2][5][2]={10,20,30,40,50,60,70,
                        80,90,100,18,21,3,4,
                        5,6,7,81,9,11};
    int (*p)[5][2];
    p=array;
    for( int i=0; i<2 ;i++)
        cout<<"\nthe vale is "<<*((int*)(p+1) + (1*2) + i);
    return 0;
}
```

```
int main() { 21
    int number1 = 88, number2 = 22;
    int* pNumber1 = &number1;
    *pNumber1 = 99;
    cout << *pNumber1 << endl;
    cout << &number1 << endl;
    cout << pNumber1 << endl;
    cout << &pNumber1 << endl;
    pNumber1 = &number2;
    int& refNumber1 = number1;
    refNumber1 = 11;
    cout << refNumber1 << endl;
    cout << &number1 << endl;
    cout << &refNumber1 << endl;
    refNumber1 = number2;
    number2++;
    cout << refNumber1 << endl;
    cout << number1 << endl;
    cout << number2 << endl;
    return 0;
}
```

Output/Error:

Segmentation fault
 Runtime Error
 Accessing The element
 That is not present
 in array.

Output:

(write "Address of **** Variable"
 where you identify the program will
 print address instead of writing some
 hypothetical address. **** will be
 replaced by the name of the variable
 whose address is assumed to be
 printed.)

99 ✓
 Address of number1 variable
 Address of number1 variable
 Address of pNumber1 variable

11
 Address of number1 variable
 Address of number1 variable

23
 11
 23 ✓

U

```
int print_row(int ct, int num)
{
    if (num == 0)
        return ct;
    cout << ct << "\t";
    print_row(ct + 1, num - 1);
}
void pattern(int n, int count, int num)
{
    if (n == 0)
        return;
    count = print_row(count, num);
    cout << endl;
    pattern(n - 1, count, num + 1);
}
int main()
{
    int n = 5;
    pattern(n, 1, 1);
    return 0;
}
```

Output/Error:

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5



Question 02 [10 marks]

Complete the following program as mentioned in the instructions and print the output as required.

```
int main() {
    // use the following dimensions of an array
    int X = 4;
    int Y = 4;
    int Z = 5;
```

Write code for dynamic allocation of a 3D array named A, using a triple pointer [2 marks]

```
int ***A = new int **[X];
for (int i=0; i<X; i++)
{
    A[i] = new int *[Y];
    for (int j=0; j<Y; j++)
    {
        A[i][j] = new int [Z];
    }
}
```

Assume that you have populated the array as following at each index using formula $i+j+k$; write the line within the given nested loop structure to assign those values to each index, using pointer notations.

[1 Marks]

```
for (int i = 0; i < X; ++i){
    for (int j = 0; j < Y; ++j){
        for (int k = 0; k < Z; ++k){
```

$$\star(\star(\star(A+i)+j)+k) = i+j+k;$$

```
}
```

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

}

Use pointer arithmetic and print the output of the following statements. [5 marks]

Statement	Output
<code>cout << ***A << endl;</code>	Q ✓
<code>cout << *(*(*(A+1) + 2)) + 1 << endl;</code>	4 ✓
<code>cout << *(*(*(A+1)+2)+2) + 100 << endl;</code>	108 ✓
<code>cout << *(*(*(A + 2) + 2)) + 4 << endl;</code>	8 ✓
<code>cout << * (*(*(A + 2)))<< endl;</code>	2 ✓

write your code to De-allocate the dynamic memory of the 3D array A [2 marks]

```

for(int i=0; i< X; ++i) {
    for(int j=0; j< Y; ++j) {
        delete []A[i][j];
    }
    delete []A[i];
}
delete []A;
}

return 0;
}

```

Question 03 [10 marks]

Complete the following C++ code that checks if a given char array has balanced parentheses or not.

Example01:

Input string: "00"
Output: true

Example02:

Input string: "(0)0"
Output: true

Example03:

Input string: "(00"
Output: false

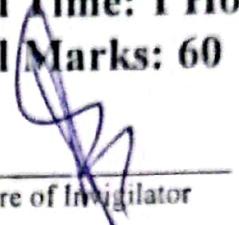
Example04:

Input string: ")"
Output: false

```
#include <iostream>
using namespace std;
bool isbalancedparentheses(char* s, int length, int open_count = 0) {
    if (open_count < 0) {
        return false;
    }
    if (length == 0) {
        return _____; ✓
    }
    if (_____ == '(') {
        return isbalancedparentheses(s + 1, length - 1, open_count + 1);
    }
    else if (_____ == ')') {
        return isbalancedparentheses(s + 1, length - 1, _____); ✓
    }
    else {
        return isbalancedparentheses(s + 1, length - 1, _____); ✓
    }
}
int main() {
    char s[] = "()()"; // Input Example string to check
    int length = sizeof(s) - 1; // Get the length of the string 14
    if (isbalancedparentheses(s, length)) {
        cout << "The string is balanced." << endl;
    }
    else {
        cout << "The string is not balanced." << endl;
    }
    return 0;
}
```

CS-1004: Object Oriented Programming (CS)

Serial No:
Sessional Exam-II
Total Time: 1 Hour
Total Marks: 60


 Signature of Invigilator

Monday, 10th April, 2023

Course Instructors

Amna Irum, Shams Farooq, Shehreyar Rashid,
 Mariam Hida

Aneeq Ahmed M. Iftikhar
 Student Name

Roll No.

CS-B
 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 ten (10) different printed pages including this title page. There are a total of 2 questions. And write your instructor's complete office number in front of point 6 to secure bonus marks.
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.

C-20SF

	Q-1	Q-2	Total
Marks Obtained	30	20	50
Total Marks	30	30	60

+ 5 = 55

V good

Question 01 [30 marks]

What would be the output produced by executing the following C++ codes? Identify errors, if any (either write output or error (syntax/runtime), both will not be accepted). All the code snippet contains #include<iostream> and using namespace std;

A [3 Marks]

```
class Box{
    int capacity;
    bool operator<(Box b){
        return this->capacity < b.capacity ? true : false;
    }
public:
    Box(){}
    Box(double capacity){ this->capacity = capacity; }
};
int main(){
    Box b1(10);
    Box b2(14);
    if(b1 < b2){
        cout<<"Box 2 has large capacity.";
    }
    else{
        cout<<"Box 1 has large capacity.";
    }
    return 0;
}
```

Output/Error:

Error

bcz overloaded(<) is private, so not accessible inside main().



B [5 Marks]

```
class Point{
    int x, y;
    public:
        Point(int x=0, int y=0){
            this->x=x; Point::y=y;
            (*this)();
        }
        void operator()(){
            cout<< " ("<<x<<","<<y<<") " <<endl;
        }
        Point& operator()(int y){
            this->y=y;
            return *this;
        }
        ~Point(){
            cout<<"Point is going"; (*this)();
        }
} p3;
int main() {
    Point *p=new Point(5,6);
    static Point p1(p3);
    p1(9)(8);
    delete p;
    Point p2(7);
    cout<<"-----" <<endl;
    return 0;
}
```

Output/Error:

~~(0,0)~~

(0,0) ✓

(5,6) ✓

Point is going (5,6)

(7,0) ✓

Point is going (7,0)

Point is going (0,8)

Point is going (0,0)



National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

C [7 Marks]

```

class Complex{
    double r, i;
public:
Complex(double r=1.0,double i=1.0){
    set(r,i);
}
void set(double r,double i){
    Complex::r = r; this->i = i;
}
void print(){
    if (i<0)
        cout << r << " " << i <<"i" << endl;
    else
        cout << r << "+" << i <<"i" << endl;
}
Complex operator+(Complex R){
    Complex tmp;
    tmp.r = r + R.r;
    tmp.i = i + R.i;
    return tmp;
}
Complex operator++(){
    Complex tmp=*this;
    r++; i++;
    return tmp;
}
Complex operator++(int){
    ++(*this);
    return *this;
}
};
int main()
{
    Complex A(3,4), B(5,-6);
    A.print();
    B.print();
    Complex C;
    C = A+B;
    C.print();
    (++A).print();
    C = ++A;
    C.print();
    (A++) .print();
    A.print();
    return 0;
}

```

Output/Error:

$3+4i$ ✓
 $5-6i$ ✓
 $8-2i$ ✓
 $3+4i$ ✓
 $4+5i$ ✓
 $6+7i$ ✓
 $6+7i$ ✓

7

D [5 Marks]

```

class Mystery {
public:
static int n;
Mystery() {cout << n++ << endl; }
Mystery(int i) {n=i;cout << n << endl;}
static void somefunc(){ n=5; }

Mystery(Mystery const& otherNum){
    n+=5;
    cout << n << endl;
}
~Mystery(){cout << --n << " ";}
} a; ✓

void fun(Mystery n){
    cout << n.n << endl;
    n.somefunc();
}

int Mystery::n=0;

int main(){
Mystery b(9), c;
fun(b);
return 0;
}

```

Output/Error:

0 ✓
9 ✓
9 ✓
15 ✓
15 ✓
4 3 2 1

5

National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2023

Islamabad Campus

E [10 Marks]

```
class ZooCage
{
    int cageNumber;
    ZooCage* link;
public:
    ZooCage(int n):cageNumber(n),link(NULL){}
    int getCageNumber() {return cageNumber;}
    ZooCage*& getLink() {return link;}
}*start=NULL;
void fun(ZooCage *& H, int num)
{
    if(H){
        fun(H->getLink(),num);
        return;
    }
    H = new ZooCage(num);
}
void fun(ZooCage * H)
{
    if(H){
        fun(H->getLink());
        cout<<H->getCageNumber()<<endl;
    }
}
int main()
{
    fun(start,4);
    fun(start,2);
    ZooCage * temp=new ZooCage(5);
    temp->getLink()=start->getLink();
    start->getLink()=temp;
    fun(start,3);
    fun(start);
    return 0;
}
```

Output/Error:

3
2
5
4

✓
✓
✓
✓

10

Question 02 [30 marks]

Your goal is to write a program for creating a MovieStore. Your MovieStore should allow for storage of many movie names, there (**unique**) IDs and ratings. Your MovieStore should allow facility of performing following operations, (minimum) data members required are given (you can add more). You need to add the necessary functions as per sequence, following the sequence will give you 2 marks:

```

int main(){
    MovieStore s1, s2(10); // create two stores of size 5 and 10

    s1["Hobbit"] = 4.5;
/* If previously not added, add a movie with average rating of 4.5
If already exist, overwrite rating by 4.5
If memory finished print "Memory finished overwriting rating of last movie"
And Update accordingly */

    s1["Lord of the Rings"] = 5;

    cout << s1[2] << endl;
/* get rating for movie id 2 (Lord of the Rings => 5).
If id not found print "id not found"
And return -1 rating.*/
    s2=s1; // copies elements from s1 to s2

    s1["Lion King"] = 8;
    cout << s2; // should display all the Movie store information
return 0;
}

```

Expected Output:

```

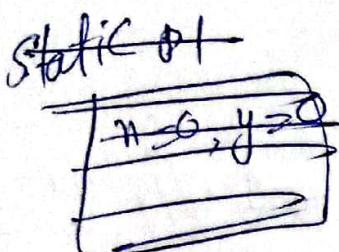
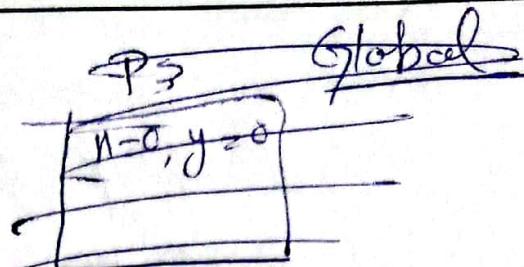
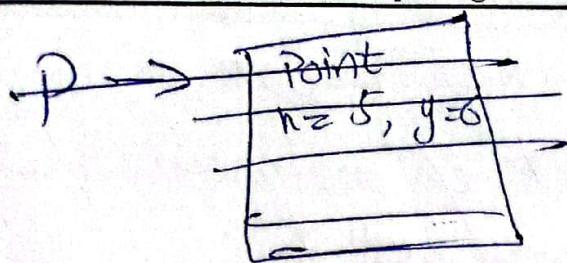
5
MovieStore:
Id   Movie           Rating
1    Hobbit          4.5
2    Lord of the Rings  5

```

```

class MovieStore{
    string * names;
    int * ids;
    float * rating;
    int size;
    int currentSize;
    .....
}

```



P'

```
public:
MovieStore (int size = 5)
{
    this->size = size;
    this->names = new string [size];
    this->ids = new int [size];
    this->rating = new float [size];
}
```

3

```
float & operator [] (string name)
{
    bool flag = false;
    for (int i = 0; i < this->currentSize; i++)
    {
        if (this->names[i] == name)
        {
            return this->rating[i];
        }
    }
}
```

12
8
20

(5)

```
this->names[currentSize] = name;
if (this->currentSize == this->size - 1)
{
    cout << "Memory finished overwriting
            rating of lost movie";
    this->names[currentSize] = name;
}
```

```

    return this->rating [currentSize];
}

// End of if to check if memory is full
// Control reaches here if memory not full
this->names [currentSize] = name;
this->currentSize++;
return this->rating [currentSize-1]; // return currentSize-1
}

// End of function

```

```

float operator [] (float int number)
{
    if (this->currentSize + 1 <= number)
        cout << "id not found";
    return -1;
}

// if id exceeds currentSize
return this->rating [number-1];

```

3

~~MoviesStore (const & MoviesStore & M)~~

```

this->size = M.size;
this->currentSize = M.currentSize;
for (int i=0; i<
    this->names = new string [this->size];
    this->ids = new int [this->size];

```

~~this->rating = new float [this->size],
for (int i=0; i<this->currentSize; i++)
{ this->names[i] = M.names[i];
this->ratey[i] = M.rating[i];
this->ids[i] = M.ids[i];
}~~

~~3 // End of function~~

~~~MovieStore()~~

~~{ delete [] names,  
delete [] ids;  
delete [] rateys;  
}~~

(3)

~~MovieStore operator=(const & MovieStore m)~~

~~{ this->size = M.size; this->currentSize = M.currentSize;~~  
~~this->names = new string [size],  
this->ids = new int [size],  
this->ratey = new float [size],  
for (int i=0, i<currentSize; i++)  
{ this->names[i] = M.names[i],  
this->ratey[i] = M.rating[i],  
this->ids[i] = M.ids[i];  
}~~

3; 3

# CS1004 Object Oriented Programming

Saturday April 08, 2023

# Course Instructor

Mr. Rizwan Ul Haq, Mr. Ch. Usman Ghous, Dr. Bilal Khan, Dr. Imran Babar, Dr. Khalid Hussain, Mr. Hafiz Saud

**Serial No:**  
**2<sup>nd</sup> Mid Term Exam**  
**Total Time:1 Hour**  
**Total Marks: 45**

### Signature of Invigilator

Roll No \_\_\_\_\_ Section \_\_\_\_\_ Signature \_\_\_\_\_.

**DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.**

## Instructions:

1. Verify at the start of the exam that you have a total of two (03) questions printed on eight (08) pages including this title page.
  2. Attempt all questions in a space given against each question in the given order clearly. *Cutting and overwriting will not be graded.*
  3. The exam is closed books, closed notes. Please see that the area in your threshold is free of any material classified as ‘useful in the paper’ or else there may a charge of cheating.
  4. Read the questions carefully for clarity of context and understanding of meaning and make assumptions wherever required, for neither the invigilator will address your queries, nor the teacher/examiner will come to the examination hall for any assistance.
  5. Fit in all your answers in the provided space. You may use extra space on the last page if required. If you do so, clearly mark question/part number on that page to avoid confusion.
  6. Use only permanent ink-pens. Only the questions attempted with permanent ink-pens will be considered. Any part of paper done in lead pencil cannot be claimed for checking/rechecking.

|                       | Q-1 | Q-2 | Q-3 | Total     |
|-----------------------|-----|-----|-----|-----------|
| <b>Total Marks</b>    | 15  | 15  | 15  | <b>45</b> |
| <b>Marks Obtained</b> |     |     |     |           |

**Vetted By:** \_\_\_\_\_ **Vetter Signature:** \_\_\_\_\_

## **University Answer Sheet Required:**

**No**

**Yes**

# National University of Computer and Emerging Sciences

FAST NU Department of Computer Science

CFD Campus

**Q1. CLO-2**

**5+2+2+3+3 = 15 Marks**

Consider a student class with id and gpa as int and float data members respectively. The class has default constructor (default value of id and gpa are 123 and 2.5 respectively) and other three methods, **setID**, **setGPA**, and **show** functions to set id gpa and display them. In the main, call all class three methods **using single statement**. The methods **prototype must** be in such a way that all methods can be called in a chain. The class must **restrict its data members** from outside access.

- (a) Write the class definition with proper access specifiers and prototypes to claim any mark [ data members and class body 1.5 marks, default constructor 1 marks, methods prototype and definition 2.5 marks, total 5 marks].

```
class student{
    private:
        int id;
        float gpa;
    public:
        student(int id=123, float gpa=2.5){
            this->id=id;
            this->gpa=gpa;
        }
        student& setID(int id){
            this->id=id;
            return *this;
        }
        student& setGPA(float gpa){
            this->gpa=gpa;
            return *this;
        }
        void display()
        {
            cout<<"ID is :"<<id<<"\t";
            cout<<"GPA is :"<<gpa<<endl;
        }
};
```

- (b) Driver program, create an object of student and then call all setID and setGPA and then display them by calling display method. Calling **must be in a chain** and in **single statement**, otherwise zero marks [ 2 Marks].

```
int main(){

    student std;
    std.setID(957).setGPA(2.5).display();

    return 0;
}
```

- (c) Now we want to create 100 students objects and want to keep track that how many students objects are created immediately after creating objects using another class variable **count**, the count should be of such type that it can be called even when no object is created. What

# National University of Computer and Emerging Sciences

FAST NU Department of Computer Science

CFD Campus

else code need to be inserted in the existing class definition that guarantee the correct number of student objects created at that time, without any explicit function call [ 1 + 1 = 2 marks]

```
// code to add data member count

private: static int count
//code to update count immediately after every object creation
student(int id=123, float gpa=2.5){
    this->id=id;
    this->gpa=gpa;
count++; // count++ will be inserted into constructor body
}
// if someone writes: insertion of count++ static variable into default constructor of student class it's also okay
```

- (d) Now suppose, there are two instances of student class obj1 and obj2, obj2 needs to replica of obj1 i.e. obj2(obj1), **which constructor** will be called for this replication, write the constructor **definition** with proper argument [1 + 2 = 3 marks]

copy constructor will be called

```
student(student& obj){
    this->id=obj.id;
    this->gpa=obj.gpa;
}
```

- (e) Suppose we have a class defined below, write the output of this program [3 marks]

```
#include <iostream>
using namespace std;
class student{
private:
    int id;
    float *gpa;

public:
    student(int id=222, float gpa=2.5){
        this->id=id;
        this->gpa=new float;
        *(this->gpa)=gpa;
    }

    void updateGPA(float GPA)
    {
        *gpa=GPA;
    }
}
```

```
student(student& s){
    this->id=s.id;
    this->gpa=s.gpa;
};

int main()
{
    student s1;
    student s2=s1;
    s2.updateGPA(3.5);
    cout<<s2.getGPA()<<endl;
    cout<<s1.getGPA()<<endl;
    return 0;
}

What will be the output:?
3.5
3.5
```

# National University of Computer and Emerging Sciences

FAST NU Department of Computer Science

CFD Campus

**Q2. CLO-2**

**2 + 2 + 2 + 5 + 4 = 15 Marks**

Assume a Teacher and a Supervisor class is directly derived from the Employee class. An Employee class has an employee\_id of type int and an employee\_name of type string as private members. Similarly, a Teacher class has subject\_name of type string as a private member. Also Supervisor has total\_student of type int as private members. All the three classes provide getter functions to return the values of their members if the getter function is invoked on its corresponding objects.

You are required to write a program for the below:

- (a) A complete Employee class with required constructor(s) and the getter function. [2 marks]

```
class employee {  
public:  
employee(string _name="", int  
_id=0): name(_name), id(_id)  
{ }  
void getEmployee() {  
    cout<< id<<" " << name; }  
private:  
  
int id;  
string name;  
};
```

- (b) A complete Teacher class with required constructor(s) and the getter function. [2 marks]

```
class Teacher: public virtual  
employee  
{  
public:  
    Teacher(string sub) :  
    subject(sub) {}  
  
    void getTeacher()  
    {cout<<subject; }  
  
private:  
    string subject;  
};
```

# National University of Computer and Emerging Sciences

FAST NU Department of Computer Science

CFD Campus

- (c) A complete Supervisor class with the required constructor(s) and the getter function [2 marks]

```
class Supervisor: public  
virtual employee  
{  
public:  
    Supervisor(int  
total_st):students(total_st){  
    }  
    void getSupervisor()  
{cout<<students; }  
  
private:  
int students;  
};
```

- (d) A class named Derived, which is derived directly from classes Teacher and Supervisor. It provides constructor(s) and a single getter function, which will display all the information of the base classes from which it is directly and indirectly derived. [5 marks].

```
class Derived:Teacher,Supervisor {  
public:  
    Derived(int total_st, string _sub, int  
_id, string _name): employee(_name,_id),  
Supervisor(total_st), Teacher(_sub) { }  
void getDerived() {  
  
    cout<<"\nEmployee ID and name: ";  
    getEmployee();  
    cout<<"\n Teacher Subjects: ";  
    getTeacher();  
    cout<<"\n Supervisor Students: ";  
    getSupervisor();  
}  
  
private:  
};
```

# National University of Computer and Emerging Sciences

## FAST NU Department of Computer Science

## CFD Campus

- (e) In main() function, (i) create a static array of 5 Derived class objects. Furthermore, you will have to provide all the necessary information to the objects at the time of their creation. There is no need to provide information via console. In fact, there is no setter function in anyone of the classes. (ii) Use the getter function provided by the Derived class to display all the information in a descent way. [2+2 marks]

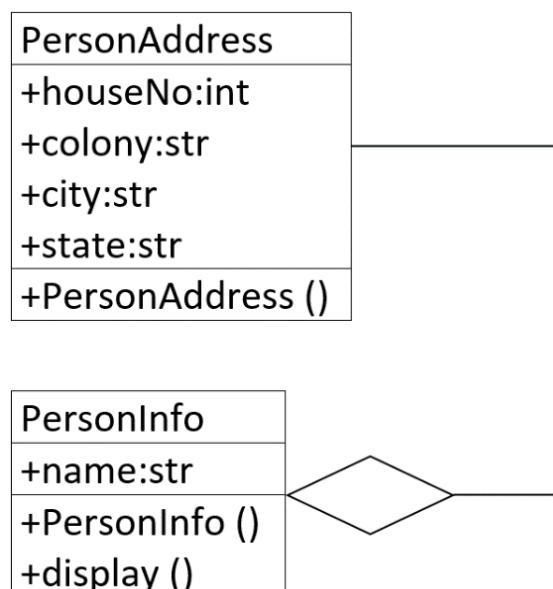
```
int main() {
    Derived d1[5] = {{Derived(10,
    "MATHS", 1, "Hanif")},{Derived(11,
    "ENGLISH", 2, "Rizwan")},{Derived(12,
    "OOP", 3, "Asad")},{Derived(13,
    "CALCULUS", 4, "Saba")},{Derived(14,
    "Algebra", 5, "Sarah")}};

    for (int i=0;i<5;i++) {
        d1[i].getDerived();
    }
}
```

**Q3. CLO-3**

**8 + 7 = 15 Marks**

- a) Implement the given aggregation scenario for the class diagram as shown below. The code must map with the class diagram. **Marks: [data member (2) + member functions (3) + aggregation (3)]**

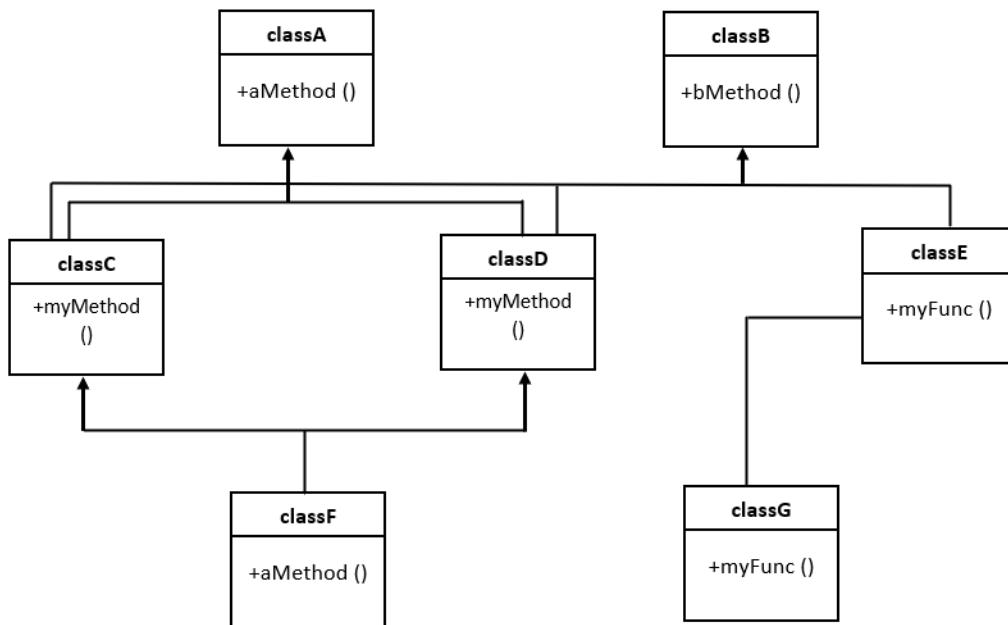


```

class PersonAddress{
public:
int houseNo;
string colony, city, state;      //MARKS[1]
PersonAddress(int hno, string colony, string
city, string state){
this->houseNO=hno;
this->colony=colony;
this->city=city;
this->state=state;           //MARKS[1]
};
class PersonInfo{
private:
PersonAddress *address;      //MARKS[1]
public:
string name;
PersonInfo(string name, PersonAddress
*address){
this->name=name;
this->address=address;}      //MARKS[1]
void
display(){cout<<name<<address-
>houseNo<<address->colony<<address-
>city<<address->state;}      //MARKS[1]
};
int main()                  //MARKS[3]
PersonAddress
objAdd=PersonAddress("parameters");
PersonInfo p1=Person("Imran's Address:",&objAdd);
PersonInfo p2=Person("Ali's Address:",&objAdd);
p1.display();
p2.display();
return 0;
}

```

- b) Draw the code structure for the following class diagram. Call methods of base classes in child classes by keeping in view the concepts of inheritance, association and method overriding. [Marks: 7]



```

class classA{
public:
aMethod();
};

class classB{
public:
bMethod();
};

class classC: public classA, public classB{
public:
myMethod();
};

class classC: public classA, public classB{
public:
myMethod();
};

class classF: public classC, public classD{
public:
aMethod();
};

class classE: public classB{
public:
myFunc();
};
  
```

```
class classG{  
    classE objclassE;           //Association  
public:  
    myFunc();  
};                                //MARKS [2]  
int main(){  
    //method calling through class objects      //MARKS[1]  
    classF objclassF;  
    objclassF.aMethod();                  //method of classF will be called  
    classA objclassA;  
    objclassA.aMethod();                  //method of classA will be called  
  
    //Method Overriding                      //MARKS [2]  
    classA *objclassA;  
    objclassA = &objclassF;  
    objclassA->aMethod();                //method of classA will be called as the pointer is of  
    the  
    //type classA  
    objclassA->aMethod();                //method of classF will be called if base class  
    aMethod is  
    //virtual  
    //Diamond Problem Solution             //MARKS[2]  
    objclassF.classC::aMethod();          //Diamond problem solution through path selection,  
    method  
    //of classA will be called  
    objclassF.classD::aMethod();          //Diamond problem solution through path selection,  
    method  
    //of classA will be called  
    classG objclassG;  
    objclassG.myFunc();                  //Method of classG will be called  
}  
  
If a student has used disinheritance method or path selection method for diamond problem  
the marks will be 2 in both of the cases.  
//Concept of Disinheriatance to resolve diamond problem keeping in view the methods are  
still overridden  
class classF: virtual public classC, virtual public classD{  
public:  
    aMethod();  
};
```

# CS-1004 Object Oriented Programming BS(DS) BS(AI)

Monday, February 27, 2023

## Course Instructors

Dr. Ishtiaq, Hassan Raza, Adil Majeed

Serial No:  
**Mid-I Exam**  
**Total Time: 1 Hour**  
**Total Marks: 80**

---

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. Last pages are available for rough work.
3. If you need more space write on the last paper and clearly mark question and part number etc.
4. After asked to commence the exam, please verify that you have (12) different printed pages including this title page. There are total of (3) 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 read the question thoroughly and use your time **properly**, an uneven distribution of time can lead to incomplete answers.

|                       | I  | II | III | Total |
|-----------------------|----|----|-----|-------|
| <b>Total Marks</b>    | 50 | 20 | 10  | 80    |
| <b>Marks Obtained</b> |    |    |     |       |

**Question I.....(50 Marks)**

Answer the following questions. In case of output/dry-run, only the code of main function is written without return 0 statement, So please don't identify these errors. Write answers on the line(s) given at the end of each part.

**Assumptions for dry-run or output:**

- uninitialized variable has value zero in beginning.
- an integer variable takes 4 bytes, float 4 bytes, double 8 bytes and char takes 1 byte.

(a) **(1 Mark)** How much memory a pointer to character variable takes?

---



---

(b) **(2 Marks)** What is the output of the following program segment? Identify errors (if any). Assume starting address of array is 0xFF10.

```

1 char* p1="Hello World";
2 int * p2;
3 cout << sizeof(p1)-sizeof(p2);

```

---



---

(c) **(2 Marks)** What is the output of the following program segment? Identify errors (if any). Assume starting address of array is 0xFF20.

```

1 char* p1="Hello World";
2 cout << &p1[5]-p1;

```

---

(d) **(2 Marks)** What is the output of the following program segment? Identify errors (if any). Assume starting address of array is 0xFF30.

```

1 int a = 57;
2 int* ptr = &a;
3 int*&pptr = ptr;
4 cout<<*pptr;

```

---

(e) **(2 Marks)** What is the output of the following program segment? Identify errors (if any). Assume starting address of array is 0xFF40.

```

1 int arr[5]={5,2,6,4,7};
2 int *p = arr + 1;
3 int *p1 = p + 3;
4 cout<<p - p1<<endl;

```

---



---

(f) **(3 Marks)** What is the output of the following program segment? Identify errors (if any).

```

1 int check[26]={0};
2 const char* word =new char[20];
3 word ="MaiPakistanHun";
4 int i=0;
5 while(word[i]!='\0')
{
    check[int(word[i]-'a')]++;
    i++;
}

```

```

9   }
10  for (int i=0; i<26; i++)
11  {
12      cout<<char(i+97)<<" :"<<check[i]<< " ";
13  }

```

---



---



---

- (g) (3 Marks) What is the output of the following program segment? Identify errors (if any).

```

1  char** d_p[] = {str+3, str+2, str+1, str};
2  char*** t_ptr[] = {d_p, d_p+1};
3  char**** f_ptr;
4  cout<<"1\n";
5  for (int i=0; i<4; i++)
6      cout<<***(d_p+i)<<"\n";
7  cout<<"2\n";
8  cout<<***t_ptr<<"\n"<<*** (t_ptr+1)<<"\n";
9  cout<<"3\n";
10 f_ptr = t_ptr;
11 cout<<***f_ptr<<"\n"<<*** (f_ptr+1)<<"\n";

```

---



---



---

- (h) (3 Marks) What is the output of the following program segment? Identify errors (if any).

```

1  bool test(int n)
2  {
3      if (n < 4)
4      {
5          test(test(++n));
6      }
7      cout<<n<<endl;
8      return n;
9  }
10 int main()
11 {
12     cout << test(2);
13     return 0;
14 }

```

---



---



---

- (i) (4 Marks) What is the output of the following program segment? Identify errors (if any).

```

1  #include <iostream>
2  using namespace std;
3

```

```
4  bool status(char* str, int st, int en) {
5      if(st>=en) return true;
6      if(str[st] != str[en]) return false;
7      return status(str, st+1, en-1);
8  }
9  int main(){
10     char * str = "racecar";
11     int s = 7 ;
12     if(status(str,0,s-1))
13         cout << "Status true";
14     else
15         cout << "Status false";
16     return 0;
17 }
```

---

- (j) (3 Marks) What is the output of the following program segment? Identify errors (if any).

```
1  #include <iostream>
2  using namespace std;
3
4  void inc(int *p) {
5      (*p)++;
6  }
7
8  int main() {
9      int a[] = {1, 2, 3, 4, 5};
10     int *p = a;
11     inc(p++);
12     cout << *p << endl;
13     return 0;
14 }
```

---

- (k) (3 Marks) What is the output of the following program segment? Identify errors (if any).

```
1  #include <iostream>
2  using namespace std ;
3  int main() {
4      int x = 5;
5      int *p = &x;
6      int **q = &p;
7      int ***r = &q;
8      cout << ***r << endl;
9      ***r = 10;
10     cout << x << endl;
11     return 0;
12 }
```

---

---

- (l) (3 Marks) What is the output of the following program segment? Identify errors (if any).

```
1 #include <iostream>
2
3 int main() {
4     int a[] = {1, 2, 3, 4, 5};
5     int *p = a + 2;
6     int *q = a;
7     *(q++) = *(--p);
8     cout << *(++p) << endl;
9     cout << *(++q) << endl;
10    return 0;
11 }
```

---

---

---

- (m) (2 Marks) What is the output of the following program segment? Identify errors (if any). Assume starting address of array is 0xFF10.

```
1 char p1[]="Hello World";
2 for(int i=0; i<5;i++)
3 {
4     p1++;
5     cout << *p1;
6 }
```

---

- (n) (2 Marks) What is the output of the following program segment? Identify errors (if any). Assume starting address of array is 0xFF10.

```
1 int p1[]={5,9,12,14,72};
2 for(int i=0; i<5;i++)
3     cout << *p1+i<< " ";
4
```

---

- (o) (2 Marks) What is the output of the following program segment? Identify errors (if any). Assume starting address of array is 0xFF10.

```
1 int arr[]={5,9,12,14,72};
2 int *p2=arr;
3 for(int i=0; i<5;i++) {
4     p2++;
5     cout << *p2+i<< " ";
6 }
```

---

- (p) (2 Marks) What is the output of the following program segment? Identify errors (if any). Assume starting address of array is 0xFF10.

```
1 int arr[]={5,9,12,14,72};
2 int *p2=arr+5;
3 for(int i=0; i<5;i++) {
4     p2--;
5     cout << *p2+i<< " ";
6 }
```

- (q) **(2 Marks)** What is the output of the following program segment? Identify errors (if any). Assume address of x variable is 0xFF50 and address of p2 is 0xFF70.

```
1 int x=5;
2 int *p2=&x;
3 cout << (*&p2);
```

---

- (r) **(1 Mark)** What is the output of the following program segment? Identify errors (if any). Assume address of x variable is 0xFF50 and address of p2 is 0xFF70.

```
1 int x=5;
2 int *p2=&x;
3 cout << (&p2);
```

---

- (s) **(1 Mark)** What is the output of the following program segment? Identify errors (if any). Assume address of x variable is 0xFF50 and address of p2 is 0xFF70.

```
1 int x=5;
2 int *p2=&x;
3 cout << (&*&*&x);
```

---

- (t) **(1 Mark)** What is the output of the following program segment? Identify errors (if any). Assume address of x variable is 0xFF50 and address of p2 is 0xFF70.

```
1 int x=5;
2 int *p2=&x;
3 cout << (*&*&*&x);
```

---

- (u) **(1 Mark)** What is the output of the following program segment? Identify errors (if any). Assume address of x variable is 0xFF50 and address of p2 is 0xFF70.

```
1 int x=5;
2 int *p2=&x;
3 cout << (*&*&x);
```

---

- (v) **(1 Mark)** For the code given below

```
1 int a=5, b=9;
2 const int *p;
3 p = &a;
```

Is it necessary to initialize p at the time of declaration? Has this code error?

Yes

No

- (w) **(1 Mark)** For the code given below

```
1 int a=5, b=9;
2 const int *p = &a;
```

Which of the following statements are not allowed for this declaration of pointer.

\*p=12;

- p=&b;*
- All of the above*
- None of the above*

(x) **(1 Mark)** For the code given below

```
1 int a=5, b=9;  
2 int * const p = &a;
```

Which of the following statements are not allowed for this declaration of pointer.

- \*p=12;*
- p=&b;*
- All of the above*
- None of the above*

(y) **(1 Mark)** For the code given below

```
1 int a=5, b=9;  
2 const int * const p = &a;
```

Which of the following statements are not allowed for this declaration of pointer.

- \*p=12;*
- p=&b;*
- All of the above*
- None of the above*

(z) **(1 Mark)** For the code given below

```
1 int a=5, b=9;  
2 int *p = &a;
```

Which of the following statements are not allowed for this declaration of pointer.

- \*p=12;*
- p=&b;*
- All of the above*
- None of the above*

**Question II.....(20 Marks)**

- (a) **(5 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 int CTD(int, int, int);
4 void main()
5 { int n=12343, b=5;
6 cout<<CTD(n, 1, b)<<endl;
7 return 0; }
8 int CTD(int n, int t, int b)
9 { if(n > 0)
10 return (n % 10)*t + CTD(n/10, t*b, b);
11 return 0; }
```

---

---

---

---

---

---

---

---

---

---

---

---

---

---

(b) **(8 Marks)** Write a recursive function in c++ which takes an integer as argument and returns the sum of those digits which are less than 5.

`sumOfDigitsLessThan5(165482)` will return 7 because digits less than 5 are 1, 4, 2 and their sum is 7.

```
1 int sumOfDigitsLessThan5(int n){  
2 //write body of this function  
3 }
```

---

---

---

---

---

---

---

---

---

---

---

(c) **(7 Marks)** Write a recursive function `printSeries` in c++ which takes an integer(n) as argument and print the number from 1 to n.

`printSeries(10)` will print 1 2 3 4 5 6 7 8 9 10

**Question III** ..... (10 Marks)

- (a) **(5 Marks)** Suppose you are working on a project that involves developing a program for a restaurant that allows customers to place their orders. You have been asked to implement the code that handles the order processing.

The restaurant serves a variety of dishes, each with a unique name, price, and ID number. The customer can select one or more dishes and specify the quantity for each. The program should calculate the total cost of the order and display it to the customer. To implement this functionality, you decide to use arrays to store information.

For example, you may use an array to store the ids, another array to store the prices and a 2D array to store the name of the products. To store the orders received you can note the information in a 2D array named Orders by noting the product ID and quantity in an order.

You also create a function `calculateOrderTotal` that takes the required data from the above mentioned arrays as argument. The function should calculate the total cost of the order and return it.

Write the implementation of the calculateOrderTotal function using pointers. Your implementation should handle cases where the quantity of a dish ordered is zero, and should ignore those dishes in the calculation. You can assume that the arrays passed in as arguments are of equal length.

Your implementation should also handle cases where the arrays are empty or null, and should return zero in those cases.



(b) **(5 Marks)** Consider an array of integers, where each element represents a weight at a scale. You are given two pointers to the beginning and end of the array, respectively, and a target weight that you need to achieve by selecting a combination of weights. Write a recursive function in C++ that takes these pointers and the target weight as input and returns a boolean indicating whether it is possible to achieve the target weight using the given weights. For example, given the following array of weights:

[45, 35, 25, 10, 5, 2.5]

And a target weight of 100, the function should return true because it is possible to achieve a weight of 100 using 45, 25, 25, and 5. Note: You can assume that the array is sorted in descending order and that it contains at least one weight less than or equal to the target weight. You may use additional helper functions if needed.

# CS-1004: Object Oriented Programming (CS)

Serial No:

**1<sup>st</sup> Sessional Exam**

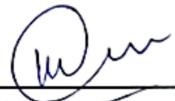
**Total Time: 1 Hour**

**Total Marks: 60**

Saturday, 24<sup>th</sup> February, 2024

## Course Instructors

Dr. Ali Zeeshan, Ms. Marium Hida, Mr. Shehreyar Rashid

  
Signature of Invigilator

Student Name \_\_\_\_\_

Roll No. \_\_\_\_\_

**DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.**

### Instructions:

1. Attempt on question paper. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work.
3. Verify that you have eight (8) different printed pages including this title page. There are two (2) questions.
4. Calculator sharing is strictly prohibited.
5. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.
6. Please consider g++ compiler for all questions in this paper.
7. Ensure that you do not have any electronic gadget (like mobile phone, smart watch, etc.) with you.

|                       | Q-1 | Q-2 | Total |
|-----------------------|-----|-----|-------|
| <b>Marks Obtained</b> |     |     |       |
| Total Marks           | 50  | 10  | 60    |

## National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2024

Islamabad Campus

## Question 1 [50 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. [2 Marks]

```
bool x = 0;
int y = 19;
char z = 's';
void* ptr = &z;
int* i = (int*)ptr;
char* c = &y;
cout << *ptr;
```

Output/Error:

Error: Void pointer is not dereferenced correctly in the cout statement.



ii. [2 Marks]

```
int var1 = 170;
int *p= &var1;
const int* ptr = p;
cout<< *p << endl;
cout<< (*p)++ << endl;
```

Output/Error:

1

iii. [2 Marks]

```
char *s[ ] = {"black", "white", "yellow", "violet"};
char **ptr[ ] = {s+3, s+2, s+1, s}, ***p;
p = ptr;
***p='a';
cout<<***p<<endl;
```

Output/Error:

# National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2024

Islamabad Campus

iv. [6 Marks]

```
char *s[] = {"black", "white", "yellow", "violet"};
char **ptr[] = {s+3, s+2, s+1, s}, ***p;
p = ptr;
cout<<**(++p)<<endl;
cout<<*(*(--(*(++p)))) + 3<<endl;
```

Output/Error:

Yellow

low

X 3

v. [6 Marks]

```
struct IntArray{
    int * arr, size;
    void create(int *ptr, int s) {
        size=s;
        arr=ptr; → [3,1,2]
    }
    void display() {
        for(int i=0;i<size;i++)
            cout<<arr[i]<<" ";
        cout<<endl;
    }
    int main() {
        int arr[]={4,0,3,1,2};
        IntArray my_arr;
        my_arr.create(arr+2,3);
        arr[my_arr.arr[0]]=arr[arr[1]];
        my_arr.display();
        return 0;
    }
}
```

Output/Error:

3 4 2

✓ 6

National University of Computer and Emerging Sciences  
FAST School of Computing Spring-2024 Islamabad Campus

vi. [6 Marks]

```
void fun(int (*p) [3]) {
    cout<<p[-1] [-4]<<endl;
}
void fun(int (*p) [2] [3]) {
    static bool flag=true;
    if(flag) {
        cout<<p[-2] [0] [7]<<endl;
        flag=false;
        fun(p - 1);
    }
    fun((int(*)[3])p);
}
int main()
{
int ary[6][3] = {{1, 2, 3}, {7,8,9}, {4,5,6}, {10,11,12}};
int (*ptr) [4]=(int(*) [4]) (ary+2); => ptr[4] => ary[2][0]
ptr--;
    fun((int(*)[2] [3]) (&ptr[2] [5]));
return 0;
}
```

Output/Error:

4.

**National University of Computer and Emerging Sciences**  
**FAST School of Computing**      **Spring-2024**      **Islamabad Campus**

vii. [6 Marks]

```

class integer{
    private:
        int i;
    public:
        integer(int ii){i=ii;}      => i = 10
        int getI(){return i;}
        void setI(int ii){i=ii;}
    };
    void display(integer i){
        cout<<"integer is "<<i.getI()<<endl;
    }
    void decrement(integer i){
        i.setI(i.getI()-1);
    }

    void increment_decrement(integer & i){
        static int s;
        if(s == 0){
            i.setI(i.getI()+1);
            s++;
        }
        else{
            decrement(i);
            s--;
        }
        display(i);
    }
int main(){
    integer i(10);
    display(i);
    increment_decrement(i);
    increment_decrement(i);
    increment_decrement(i);
    return 0;
}

```

Output/Error:

# National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2024

Islamabad Campus

viii. [5 Marks]

```

int list[5]={2,4,8,10,-1};S  

int nextList[5]={3,-1,0,1,-1};7  

int start = 2;10  

int Free = 4;2  

void magic(int val , int position){2  

    int start = ::start; =2  

    for(int i = 0 ; i< position - 1 ; i++)2  

        start=nextList[start];  

    list[Free]=val; nextList[Free]=nextList[start];  

    nextList[start]=Free++;  

}  

void magic(){  

    int start = ::start; > 2  

    while(start != -1){  

        cout<<list[start]<<"->";  

        start=nextList[start]; > 0,3,  

    }  

    cout<<"*"
}  

int main(){  

    magic();  

    magic(5,2);  

    magic();  

    return 0;  

}

```

Output/Error:

8 → 2 → 10 → 4 → \*

8 → 5 → 2 → 10 → 4 → \*

X 2.5

# National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2024

Islamabad Campus

ix. [15 Marks]

```

struct Box1 {
    int x; → 20 (one)
    Box1 *ptr1; => 83.
};

struct Box2 {
    Box1 *ptr1; → 2 (ones)
    Box2 *ptr2;
};

void print(Box2 * ptr1) {
    while(ptr1) {
        Box1* ptr2=ptr1->ptr1; →
        while(ptr2) {
            cout<<ptr2->x<<" ";
            ptr2=ptr2->ptr1;
        }
        cout<<endl;
        ptr1=ptr1->ptr2;
    }
}

int main() {
    Box1 three={10}, two={30}, one={20}, *pointer=&one; ←
    pointer->ptr1 = &three; (*(*pointer).ptr1).ptr1=&two; ←
    Box2 one1={&two}, two1={pointer} ←
    three1 = {(*pointer).ptr1, one1}, *pointer2=&three1; ←
    (*(*pointer2).ptr2).ptr2=&two1;
    print(pointer2);
    return 0;
}

```

Output/Error:

# National University of Computer and Emerging Sciences

FAST School of Computing

Spring-2024

Islamabad Campus

## Question 2 [10 Marks]

Complete the following recursive function (C++) which finds a substring in a given string. Analyze the code and write the missing statements to make the code executable.

| Example 01                                              | Example 02:                                               | Example 03                                                       |
|---------------------------------------------------------|-----------------------------------------------------------|------------------------------------------------------------------|
| text = "OOPquizspring"<br>pat = "quiz"<br>output : true | text = "OOPquizspring"<br>pat = "quiz1"<br>output : false | text = "OOPquizspring"<br>pat = "OOPquizspring"<br>output : true |

```
#include<iostream>
using namespace std;
bool exactMatch(char *text, char *pat) {

    if (_____ != _____)
        return false;

    if (*pat == '\0')
        return true;
    if (*text == *pat)

        return _____; // [2 marks]

    return _____; // [1 marks]
}

bool contains(char *text, char *pat) {

    if (*text == '\0'
        return _____; // [1 marks]

    if (*text == *pat)
        if (_____)
            return 1;
        else
            return contains(_____, _____); // [2 marks]
    }
    return contains(text + 1, pat);
}

int main() {
    cout << contains("OOPquizspring", "quiz") << endl;
    return 0;
}
```