National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Object Oriented Programming	Course Code:	CS1004
Degree Program:	BS (CS, SE, DS, BS Robotics)	Semester:	Spring 2023
Exam Duration:	180 Minutes	Total Marks:	65
Paper Date:	6-June-2023	Weight	45
Section:	ALL	Page(s):	13
Exam Type:	Final Exam		

Student : Name:______________________________ Roll No.________________ Section:_______

Instruction/Notes: Attempt all questions. Answer in the space provided. **Answers written on rough sheet will not be attached and marked**. Do not use pencil or red ink to answer the questions**.** In case of confusion or ambiguity make a reasonable assumption. Properly comment your code in Q1, Q2 and Q3.

Question 1: (CLO: 1) (Marks: 5)

Determine output for the code segment given below. There is no syntax error in the code.

```
class A{
      string myString;
public:
      A(string str = "Default"){
            myString = str;
      }
      void Print(){
          cout<<"Print A:\t"<<myString<<endl;
      }
};


class B: public A{
    C* ptr;
public:
    B(string str1 = "Default",
      string str2 = "String 2"):A(str1){
            ptr = new C(str2);
      }
      void Print(){
            A::Print();
            if(ptr != 0){
                  ptr->Print();
            }
      }
};
```

```
class C{
      string myString;
public:
      C(string str = "Default"){
         myString = str;
      }
      void Print(){
        cout<<"Print C:\t"<<myString<<endl;
      }
};


void main(){
      B abc("This is ABC", "XYZ");
      abc.Print();
}
```

OUTPUT:

```
Print A:     This is ABC
Print C:     XYZ
```

Question 2: (CLO: 4) **(Marks: 5)**

Determine output for the code given below. There is no syntax error in the code.

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

template<class myType>
class MACFilter {
    myType* data;
    int size;
public:
    MACFilter(myType* arr, int s, int* ind){
        size = s;
        data = new myType[size];
        for(int i=0; i<size; i++) {
            data[ind[i]] = arr[i];
        }
    }
    int getSize(){
        return size;
    }
    myType getData(int ind){
        return data[ind];
    }
};
template<class myType1, class myType2>
void getMACAddr(MACFilter<myType1>& t1, MACFilter<myType2>& t2)
{
    int k = t2.getSize();
    for(int i=0; i<t1.getSize(); i++) {
        cout << t1.getData(--k) << t2.getData(i) << ":";
    }
}
int main() {
    const int s = 5;
    int ind[s] = {3,0,1,4,2};
    char c1[s] = {'A','B','C','D','E'};
    MACFilter<char> obj1(c1,s,ind);

    int val[s] = {9,6,8,5,7};
    MACFilter<int> obj2(val,s,ind);
    getMACAddr(obj2,obj1);
    return 0;
}
```

OUTPUT:

```
5B:9C:7E:8A:6D:
```

Question 3: (CLO: 2) **(Marks: 15)**

You have been hired by a courier company that ships different packages all over the country. Each package must have a source location, destination location and package weight (measured in pounds). The regular shipping charges of a package is Rs. 20 per pound. Recently, the company started a new service that ships packages to the destination in lesser time. There are two types of packages that can be delivered using this service: 2-day package and urgent package. The 2-day package is charged a fixed additional amount other than the regular shipment charges and urgent package is charged an additional percentage of the regular shipment charges. The company has given you a basic driver program shown below that includes all the required functionality and classes.

Write C++ classes in a proper hierarchy which enable the driver given below to compile and produce the given output. You cannot change the driver program at all. Your code also must not have any memory leaks.

```
int main() {
        const int size = 5;
        package ** pkg = new package * [size];
        pkg[0] = new package("Lahore", "Karachi", 20, );
        pkg[1] = new TwoD_package("Lahore", Islamabad", 35, 200);
        pkg[2] = new Urgent_Package("Karachi", "Lahore",25, 10);
        pkg[3] = new TwoD_package("Karachi", "Islamabad", 30, 250);
        pkg[4] = new Urgent_Package("Karachi", "Peshawar", 40, 25);

        for (int i = 0; i < size; i++){
                cout<< “Package charges: ”<<pkg[i]->comp_charges();
                delete pkg[i];
        }
        delete [] pkg;

        return 0;
}
```

Output:

```
Package charges: 400
Package charges: 900
Package charges: 550
Package charges: 850
Package charges: 1000
```

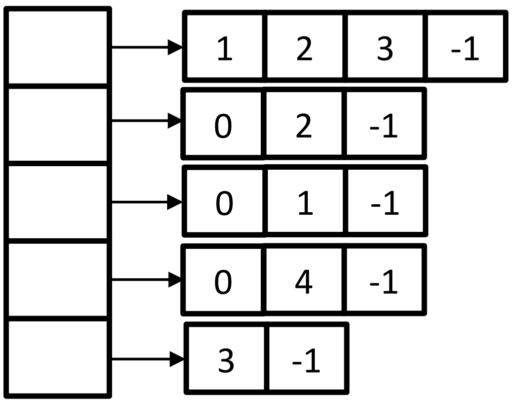
```
class package {
        string source;
        string destination;
        float weight;
public:
        package(string s, string d, float w) {
                source = s;
                destination = d;
                weight = w;
        }
        virtual float comp_charges() {
                return 20 * weight;
        }
};

class TwoD_package: public package {
        float add_charges;
public:
	TwoD_package(string s, string d, float w, float a) :package(s, d, w) {
		add_charges = a;
	}
	virtual float comp_charges() {
		return package::comp_charges() + add_charges;
	}

};

class Urgent_package :public package {
	float add_per;
public:
	Urgent_package(string s, string d, float w, float p) : package(s, d, w) {
		add_per = p;
	}
	virtual float comp_charges() {
		return package::comp_charges() * (1 +(add_per/100));
	}
};
```

Question 4: (CLO: 3) **(Marks: 15 + 15)**

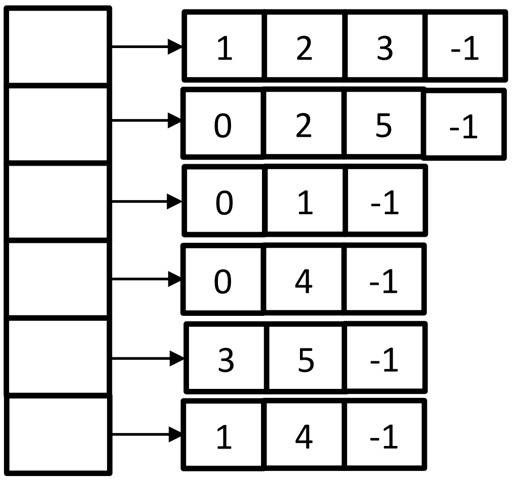
Suppose you are implementing a social network like facebook, where there are n users. Each user has certain number of friends in the network. The friends relationship is symmetric i.e. if user 1 is friend of user 2 then user 2 must also be a friend of user1. The list of friends of a user is also need to be maintained. Following class is defined for the social network.

```
class social_network{

   int num_users;// to store total number of users in the network
   int ** friends;//a dynamic 2D array to store list of friends of each user. Friends' list must end at -1

  // other members

};
```

Suppose all the users has ID from 0 to ***num_users***-1. For example consider a network of 5 users.
Where user 0 is friends with 1, 2 and 3
User 1 is friends with 0 and 2
User 2 is friends with 1 and 0
User 3 is friends with 0 and 4
And user 4 is friends with user 3 only.
The corresponding friends array will look like the following.



Part(a) Define a member function add_user that takes an integer array ***frnd*** and its size ***n*** as parameter (***frnd*** stores the ids of the friends of this new user) and add a new user with user id ***num_users*** to the network and increment the number of users by 1. It must also add its ***frnd*** information to the network. For example new user is added to the network who is friends with user 1 and 4. The friends array must be updated as follows:



Note: The dynamic memory must be allocated exactly equal to number of users and corresponding number of friends. Divide your task into smaller sub tasks and define functions for the subtasks.

```
int count_friends(int ID) {
             int i = 0;
             if(num_users>ID)
                   for (;friends[ID][i] != -1;i++);
             return i;
}

void add_Friend(int ID, int friend_ID) {
             int count = count_friends(ID);
             int* temp = new int[count + 2];
             for (int i = 0; friends[ID][i] != -1;i++) {
                   temp[i] = friends[ID][i];
             }
             temp[count] = friend_ID;
             temp[count+1] = -1;
             delete[]friends[ID];
             friends[ID] = temp;
}
void add_user(int* frnd, int size) {
             int** temp = new int* [num_users + 1];
             for (int i = 0;i < num_users;i++)
                   temp[i] = friends[i];
             delete[] friends;
             friends = temp;
             friends[num_users] = new int[size + 1];
             for (int j = 0;j < size;j++) {
                   friends[num_users][j] = frnd[j];
                   add_Friend(frnd[j], num_users);
             }
             friends[num_users][size] = -1;
             num_users++;
}
```

Part (b) One of the measures for social networks is clustering coefficient of it users. It measures how well connected a user is with his/her friends. The clustering coefficient of a user ***U*** can be computed as follows: $C_u = \frac{2e_u}{k_u(k_u-1)}$. Here $k_u$is the number of friends of user ***U*** and $e_u$is the number of pair of friends of user ***U*** that are friends themselves. For example: clustering coefficient of user 0 is $C_0 = \frac{2*1}{3(2)} = 1/3$ as user 0 has three friends but only one pair of friends (user 1 and 2) are also friends. Note that if a user has only one friend then its clustering coefficient is 0. In our example user 4 has only one friend so its clustering coefficient is 0. Write a C++ function compute_CC defined in the social network class that computes and return the clustering coefficient of each user in a dynamic array of size num_users.

```
bool is_friends(int i, int j) {
		//bool flag = false;
		for (int k = 0;friends[i][k] != -1;k++) {
			if (friends[i][k] == j)
				return true;
		}
		return false;
}
float* comp_CC() {
		float* cc = new float[num_users];
		//compute the CC of user u
		for (int u = 0;u < num_users;u++) {
			cc[u] = 0;
			int count = count_friends(u);
			int pairs = 0;
			for (int i = 0;friends[u][i] != -1;i++) {
				for (int j = i + 1;friends[u][j] != -1;j++) {
					if (is_friends(friends[u][i], friends[u][j]))
						pairs++;
				}
			}
			if (count > 1)
				cc[u] = (2.0 * pairs) / (count * (count - 1));
		}
		return cc;
}
```

Question 5: (CLO: 4) **(Marks: 10)**

Determine output for the code given below. Explicitly mention if the program crashes or leaks any resources. There is no syntax error in the code.

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

void Test1(int x);

class A {
	string myWhat;
public:
	A(string str = "Error Occured.") { myWhat = str; }
	virtual string what() { return myWhat; }
};

class B : public A {
public:
	B(string str = "Error.") :A(str) {}
};

int Test2(int x)
{
	if (x == 50)
		throw B("Exception 50");
	else if (x == 100)
		throw A("Exception 100");
	else if (x == 150) {
		string str = "Exception 150";
		throw str;
	}
	else
		return 0;
}
void SomeFunction(int x)
{
	try
	{
		Test1(x);
		cout << "Back in SomeFunction.\n";
	}
	catch (A& ex)
	{
		cout << "SomeFunction A: " << ex.what() << endl;
	}
	catch (B& ex)
	{
		cout << "SomeFunction B: " << ex.what() << endl;
	}
	catch ( string & ex)
	{
		cout << "SomeFunction Exception: " << ex << endl;
		throw ex;
	}
	cout << "- - - -\n";
}
```

```
void Test1(int x)
{
	if (x == 50)
	{
		try {
			Test2(x);
		}
```

```
            catch (string & ex) {
                  cout << "50: Test1 Exception ...\n" << ex << endl;
            }
      }
      else if (x == 100)
      {
            try {
                  Test2(x);
            }
            catch (B& ex) {
                  cout << "100: Test1 B ...\n" << ex.what() << endl;
            }
            catch (string & ex) {
                  cout << "100: Test1 Exception ...\n" << ex << endl;
            }
            catch (A& ex) {
                  cout << "100: Test1 AAA ...\n" << ex.what() << endl;
            }
      }
      else if (x == 150) {
            int* ptr = 0;
            try {
                  ptr = new int[100];
            }
            catch (B& ex) {
                  cout << "150: Test1 B ...\n" << ex.what() << endl;
            }
            catch (string & ex) {
                  cout << "150: Test1 Exception ...\n" << ex << endl;
                  throw ex;
            }
            catch (A& ex) {
                  cout << "150: Test1 AAA ...\n" << ex.what() << endl;
            }
            Test2(x);
            if (ptr)     delete[] ptr;
      }
      else
      {
            try {
                  cout << "Else ...\n";
            }
            catch (A& abc) {
                  cout << "Else: Caught A...\n" << abc.what() << endl;
            }
            catch (B& abc) {
                  cout << "Else: Caught B...\n" << abc.what() << endl;
            }
      }
      cout << "Test Printing.\n";
}
```

What will be the output for following function calls?

Part a	Part b	Part c

`void main(){`
`       SomeFunction(50);`
`}`
	`void main(){`
	`       SomeFunction(100);`
	`}`
		`void main(){`
		`       SomeFunction(150);`
		`}`

Output: **SomeFunction A: Exception 50** **- - - -**	**Output:** **100: Test1 AAA ...** **Exception 100** **Test Printing.** **Back in SomeFunction.** **- - - -**	**Output:** **SomeFunction Exception: Exception 150** **Program abort due to unhandled exception**