[1]

Problem Statement: You have provided vector of N integers, then you have given 2 problem statements

a. You have given position in the vector, this position in the vector needs to be erased
b. You have given 2 integers denoting range of the positions in the vector
   The element fall under that range should be removed

Note: The second statement is performed on the updated vector after problem statement 1 is executed

Expectation: -

1. Input the vector of N integers
2. Input position to be erased from the vector
3. Input 2 integers of position range, where the range of elements to be deleted

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;

int main()
{
        // Enter your code here
        return 0;
}
```

[2]

Problem Statement: Implementation of a different template when a specific data type is passed as template parameter

Design:

1. Create a class which store a variable of type int and a function to print it.
2. Specialized template to handle the different data type (e.g String) than the class handles.
3. Pass the arguments to different objects of the template class to achieve the runtime polymorphism

Expectations:

1. Create Application with Main function
2. Define class template to handle the data
3. Use of specialized template to handle different data types
4. Required output:
   If Int a =5 program should return a= 10
   If string abc = xyz program should return abc = xyzxyz

[3]

Problem Statement: In code, showcase use of std::function for following actions from main().

```
std::function<int(int,int)> operations[n];
operations[0] = divide();
operations[1] = multiply();
operations[2] = subtract();
operations[3] = add();
```

And call desired operation like int d = operations[0](10,5);
Divide, multiply etc should be classes with operator() overloaded.

[4]

Problem Statement: Create a product sum system that overloads the possible mathematical operators

Design:

Create a product sum system that overloads the possible mathematical operators as Given below

(a). The class should contain item name, per unit price, count of similar item, discount, total amount.

(b). The class should overload operators(+,-,++, --) such that when the two objects are added, count should be increased if similar items, amount and total amount should be adjusted as per the items.

(c). Similarly to point (b), subtraction, increment and decrement operators should be overloaded.

(d). Overload << operator such that it should print all the information of the object such as, name, count, per unit price and total etc.

Please note that different items should be different objects of the class, and it should contain a grant total amount variable(static). Add corresponding functions to support the class such as Enter product details.

Expectations:

1. Creation of class ProductSumSystem
2. Use of mathematical operator overloading ++, –-, +, - operators
3. Create Application with Main function.
4. Overload insertion operator
5. Runnable program

[5]

Problem Statement: You have two classes Prime1, Prime2 both this classes implement their own version of function “evaluate”

In class prime1, evaluateFunc multiply the value passed by 3:

```
class Prime1 {
        public:  Prime1() {varPrime1 = 0;}
        private:  int varPrime1; void calc () {varPrime1++;}
        protected:  void evaluate (int & Val) {Val = Val * 3; calc ();}
        public:      int getPrime1() {return varPrime1;}
};
```

In class prime2, evaluateFunc multiply the value passed by 5:

```
class Prime2 {
        public:  Prime2() {varPrime2 = 0;}
        private:  int varPrime2; void calc () {varPrime2++;}
        protected:  void evaluate (int & Val) {Val = Val * 5; calc ();}
        public:     int getPrime2() {return varPrime2;}
};
class PrimeImplement {
        int valPrime;
public:
        PrimeImplement() {valPrime = 1;}
        // Implement this function
        // Write code here
        void update_prime(int new_val){
        }
};
```

Expectation: You need to modify the PrimeImplement's function update_prime, this function should update PrimeImplement's valPrime variable only by calling Prime1, Prime2's evaluate function

Note: It is guaranteed that new_val of update_prime function has 3 and 5 as its prime factors

Sample Input: new_val = 15

[6]

Problem Statement: Define a class that uses a map to store phone contacts.

Design:

Write a class that uses a map to store phone contacts. Key will the contact number value will be the name(string).

(a). Using iterators check if two contact numbers have same name, if yes then print name and contact number of both.
(b). Add a function to modify contact number using iterator.
(c). Use iterator to print all the keys and values
(d). Use iterator to find and erase an item from the map

Expectations:

1. Create class PhoneContacts
2. Create Application with Main function.
3. ModifyContact function
4. Print details function
5. Find and Erase function
6. DuplicateNames function

[7]

Problem Statement: Define a Money class, suitable for storing money information.

Design:

a. Class objects should contain two int fields: euros and centimes.
b. Write for the class appropriate constructors and get/set methods to allow the user to access its data.
c. Also write a print method that prints out an object's info. Overload the + operator, so that adding two objects of this class works.
d. As an added requirement, after adding two Money objects, values of more than 100 centimes should be converted to euros.

Expectations:

1. Use Access specifiers for each attribute in class.
2. Create Application with Main function.

Print appropriate output from each function

[8]

Problem Statement: Define a Button Classes to demonstrate inheritance using Virtual Functions

Design:

Class 1: Define a Define Base class as Button, this class must have below attributes

1. Enumerations:
   a. Enumeration to list all Button states (Pressed, Released)
2. Functions
   a. Constructor
   b. Destructor
   c. Function to generate states on
   d. Handler function for each state
      i. onPressed
      ii. onReleased
   e. Virtual Method Paint to draw Button

Class 2: Create Child Class as SquareButton inherited from Button

1. Functions
   a. Implement necessary functions from Base Class to redraw button as square (Draw implementation not necessarily required, only print statement is OK)

Class 3: Create Child Class as CircularButton inherited from Button

1. Functions
   b. Implement necessary functions from Base Class to redraw button as Circular (Draw implementation not necessarily required, only print statement is OK)

Expectations:

1. Use Access specifiers for each attribute in each class.
2. Create Application with Main function.
3. Create Button Class pointer holding object for SquareButton
4. Create Button Class pointer holding object for CircularButton
5. Call ***Paint*** functions for all objects created above.

[9]

Problem Statement: Create Container class to hold Contact list and to perform operations on it. Use STL containers.

Design:

Struct 1: Create data structure with below attributes to hold Contact data

1. Variables
    a. Contact Name
    b. Contact Number
    c. Picture File Path

Class 1: Create a container Class to hold list of contact data defined in above data structure

1. Functions
    a. Add Contact to List
    b. Remove Contact from List
    c. Search Contact with Name

    [Choose appropriate Container type to provide optimal performance for above functions.]

Expectations:

3. Use Access specifiers for each attribute in each class.
4. Create Application with Main function.
5. Create Contact Container Class object in main function
6. Insert At least 5 contacts (Print list)
7. Remove any one contact (Print list)
8. Search Contact with name

[10]

Problem Statement: Define a program to add the customer details parallelly.

Design:

Blow are the customer details, Assume suitable types for each:

1. id: < id>
2. name: "user"
3. phoneNumber: "9800000000"
4. Create a method to add customer details
5. Use aync and future concept

Expectation:

1. Print the added customer details

[11]

Problem Statement: Create a user defined exception and handle it in the program, while creating an object of another class.

Design:

1. Create a class defining the exception
2. Throw the exception and catch it in driver class

Expectations:

1. Create Application with Main function
2. Instantiate another class .
3. Throw an exception if object is already created already(not required to use singleton)
4. Handle the exception thrown.

[12]

Problem Statement: Define a Class & its functionality to demonstrate usage of constexpr, auto, decltype, nullptr, static_assert etc. i.e. any 3 of the basic modern C++ constructs

Design:

Implement a MinimumQ data structure that can store comparable elements and supports the queue operations add(x), remove(), and size(), as well as the min() operation, which returns the minimum value currently stored in the data structure

Expectations:

1. Demonstrate usage of the basic modern C++ constructs like constexpr, auto, decltype, nullptr, static_assert etc.

[13]

Problem Statement: Create the program to fetch data from two source at the same time

Design:

1. Create two database having different values of same type
2. Fetch the data from both database at the same time
3. Combine both data and display it
4. Calculate the operation time

Expectation:

1. Use threading and chrono library.

[14]

Problem Statement: Write code to show use of std::bind.

Design: Class 1: In a Class provide following functions

1. divide
2. multiply
3. subtract
4. add

main 1: use std::bind to bind to above functions. Call them and print output.

[15]

Problem Statement: Define a shape Class to demonstrate override, final & inheritance.

Design:

1. Develop a class called Shape with two data members width and height.
2. The class contains member functions to input and to display the width and height.
3. Write child class Rectangle that inherits shape class. It contains a member function to calculate area of rectangle.
4. Write another child class Triangle that inherits shape class. It contains a member function to calculate area of triangle.
5. The functions to be defined are:
   - set_data
   - area

Expectations:

1. Use Access specifiers for each attribute in class.
2. Write Suitable constructors and virtual functions.
3. Demonstrate run time binding with pointers & references.
4. Create Application with Main function.
5. Use override and final keyword.

[16]

Problem Statement: Define a class to demonstrate understanding of Function Object

Design: For a game program which has number of bricks, write a program using function objects (functor) to identify all the brick object position whose color is green.

```
class Bricks
{
  public:
    Color getColor() const;
    // Write code here
};

enum class BrickColor
{
    Red,
   Green,
   Blue,
   Yellow
   // Write code here
};

Collection of Bricks
std::vector<Bricks> poolOfBricks;
// Write code here
```

Note: Request you to write all the supporting functions for positions, setting color etc. Add 20 odd Bricks elements in poolOfBricks vector which has 5 green bricks placed randomly.

Expectations:

1. Use Access specifiers for each attribute in each class.
2. Create Application with Main function.
3. Print appropriate output from each function

[17]

Problem Statement: Define a Cab management system class, details are explained as below:

Design:

Class 1: Individual classes for each entity (i.e. Employee, Supervisor, Cab, etc...) with below attributes

1. Member Variables
   a. Variable to store cab, supervisor details
2. Functions
   a. Constructor
   b. Destructor
   c. Function to accept/reject cab requests
   d. Function to get cab details

Class 2: Bookings class for cab booking

1. Member Variables
   a. Booking details, etc
2. Functions
   a. Function to add/delete cab requests
   b. Function to get booking details
   c. Function to get status

Expectations:

1. Use Access specifiers for each attribute in each class.
2. Create Application with Main function.
3. Instantiate Bookings class in main function
4. Call various Booking operations in main function to test all functions.
5. Print appropriate output from each function.

[18]

Problem Statement: Write an array class with below functionality:

Design:

a) Define an array with a maximum of n elements.

b) Implement function that take input from user like array position and its respective value.

c) If user enters invalid index then "BadIndex" error being thrown.

d) If there is no space left to insert an element in the array, an exception of "OutOfRange" should be thrown.

Expectations:

Use Access specifiers for each attribute in class.

Create Application with Main function.
Print appropriate output from each function

[19]

Problem Statement: Define a Component Classes to demonstrate inheritance using Virtual Functions

Design:

Class 1: Define Base class as Component, this class must have below attributes

1. Functions
    a. Constructor
    b. Virtual Destructor
    c. Pure Virtual method ***OnExecute***(void)
    d. Pure Virtual method ***OnMessage***(Message* msg), Define Message class with some data.

Class 2: Create Child Class as ControllerComponent inherited from Component

1. Functions
    a. Implement necessary functions from Base Class

Class 2: Create Child Class as ModelComponent inherited from Component

1. Functions
    a. Implement necessary functions from Base Class

Class 3: Create Child Class as ViewComponent inherited from Component

1. Functions
    a. Implement necessary functions from Base Class

Expectations:

9. Use Access specifiers for each attribute in each class.
10. Create Application with Main function.
11. Create Component Class pointer holding object for ControllerComponent
12. Create Component Class pointer holding object for ModelComponent
13. Create Component Class pointer holding object for ViewComponent
14. Create object for Message Class and fill with some data.
15. Call ***OnExecute*** and ***OnMessage*** functions for all objects created above.
16. Print appropriate output from each function

[20]

Problem Statement: Write a template class with below functionality:

Design:

a) Supports below data types:

I. int
ii. float
iii. char

b) This class should be able to take any of the above-mentioned data types as input and store it.

c) Also, create vector of above type and sort its elements.

Example,

Input = 5,6,4,3,6,7,2,1

Output = 1 2 3 4 5 6 6 7

Input: e a d c f b

Output: a b c d e f

Expectations:

1. Use Access specifiers for each attribute in class.
2. Create Application with Main function.
3. Print appropriate output from each function

[21]

Problem Statement: Define a Money class, suitable for storing money information.

Design:

e. Class objects should contain two int fields: euros and centimes.

f. Write for the class appropriate constructors and get/set methods to allow the user to access its data.

g. Also write a print method that prints out an object's info. Overload the + operator, so that adding two objects of this class works.

h. As an added requirement, after adding two Money objects, values of more than 100 centimes should be converted to euros.

Expectations:

Use Access specifiers for each attribute in class.

Create Application with Main function.

Print appropriate output from each function