Military Institute of Science & Technology
Department of Computer Science and Engineering
CSE 206: Object Oriented Programming Language Sessional
Practice Problem Compilation

Useful Resources

- https://www.learncpp.com/
- https://www.cplusplus.com/
- https://www.w3schools.com/cpp/
- https://www.tutorialspoint.com/cplusplus/index.htm
- https://www.programiz.com/cpp-programming
- https://www.youtube.com/watch?v=GQp1zzTwrIg (See description)

Online Compilers

- https://www.onlinegdb.com/online_c++_compiler
- https://ideone.com/
- https://rextester.com/l/cpp_online_compiler_visual
- https://www.jdoodle.com/online-compiler-c++/

1. FileHandler Class - Problem on Copy Constructor.
   Our task is to design a class that handles the file opening and closing for us.
   a. Find and print the number of vowels in a given text file.
   b. Print the content of the text file.
   Write down a class named FileHandler. It will have the following skeleton.
   ```
   class FileHandler {
         FILE * fp;
         char name[100];
   public:
   };
   ```
   The constructor will receive the filename and initialize the file pointer as well as the name (of the file). The destructor will close the file.
   Sample Code for opening a file:
   ```
   FILE * fp = fopen("input.txt", "r");
   ```
   Sample Code for closing a file:
   ```
   fclose(fp);
   ```
   The class will have two friend functions.
   a. `void printContent(FileHandler fh)`
   Prints the content of the file pointed by the filepointer in fh.
   b. void printVowelCount(FileHandler fh)
   Prints the total number of vowels in the file.
   Sample code for reading from a file:
   ```
   char line[100];              //Let, each line-length <100
   while(!feof(fp)) {
         fgets(line, 100, fp);  //read one line
         printf("%s", line);    //print the line
   }
   ```
2. Consider the following skeleton for problem a, b and c.
   ```
   class String
   {
         int len;
         char *ptr;
         public: //declare constructor, destructor and copy constructor
               char get(int ind)
   ```

```
            {
                  //return the character at the given index
                  //Check if index is out of bound
                   //return 0 in case of error
            }
            int put(int ind,char c)
            {
                  //Assign character c at position index
                   //Check if index is out of bound
                   //return -1 in case of error
            }
            int getlength()
            {
                  //return the allocation length of the string
            }
            void print()
            {
                  //print the string upto allocation size
            }

};
```

a. Design a String class that will hold the character values dynamically. Use the above mentioned skeleton. Declare a **non-member** function, which will compare the sum of ASCII characters between two strings. If the sum of ASCII characters of String 1 is greater than String 2, '1' will be printed whereas, if the sum of ASCII characters of String 2 is greater than String 1, '2' will be printed. However, for the sum of ASCII characters of two strings being the same, 0 will be printed by the program.

```
void compare(String s1,String s2)
{
      // print 1; if sum of ASCII character of s1 is greater than s2
      // print 2; if sum of ASCII character of s2 is greater than s1
      // print 0; if sum of ASCII characters of both string being
the same.
}
```

b. Declare a non-member function of the class of problem 1, which will join 2 strings and will return the resultant string.

```
String concat(String s1, String s2){
      //string s1=”abcd”
      //string 2 = “1234”
      //result= = “abcd1234”
      //Joins two strings s1 and s2 and returns the resultant
string.
}
```

c. Declare a non-member function of the class of problem 1, which will insert a particular character at a fixed position of the string and other characters of the string will be shifted to the right

```
void insert(String &st, int index, char c) {
//string s1="abcd"
//insert(s1,0,'k') //kabc
// Insert char c at index position of st. Shift other characters to
the right.}
```

3. In this problem, you'll need to design a Point class that will hold the Cartesian coordinates. Your class will have two private variables named x and y.
   a. Write down appropriate constructors and destructors.
   b. Write a member function with the following prototype:
      `Point * shiftBy(int dx, int dy)`
      [Suppose a point object is p1(3, 4). Calling p1.shiftBy(2, 6)will shift p1 to (5, 10), and return itself.]
   c. Write a non-member function to calculate the distance between two functions. It will have the following prototype.
      `double distance(Point p1, Point p2)`
   d. Write a print() function to print the coordinates.

   Requirement
   a. Declare a Point object p1 with values
   b. Declare a pointer to Point object named p2.
   c. Shift p1 by (2, 5), and hold its reference in p2.
   d. Use p2 pointer to print the value of p1
4. In this problem, you'll need to design a Stack class that will hold the integer values dynamically. Use the following skeleton of Stack class.

```
class Stack
{
      int *p; int len;
public:
      Stack(int n);//n = size of the stack
      ~Stack(){delete [ ]p;}
      initStack(int *ara)//init the stack with the values of ara
      printStack()//print the values of stack
      …//Write more methods if necessary
};
void printSum(Stack s)//non-member function, prints the sum of
elements of s.
```

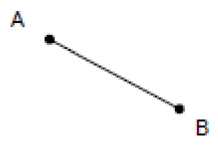
5. A complex variable has a real portion and an imaginary portion.
   a. Write down a class that represents a complex variable (all the members will be private)
   b. Write a public setter method to set the values.
   c. Write a non-member function add(…) that will take two complex Variable objects and print their summation
   d. Write a non-member function sub(…) that will take two complex Variable objects and print their subtraction.

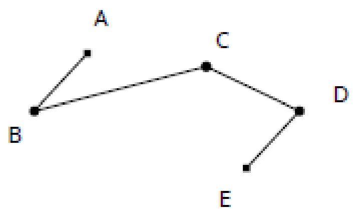
6. A Science Student of SSC has a roll number, marks in primary Subjects, and marks in Science subjects.
   A Commerce Student has a roll number, marks in primary subjects, and marks in Commerce Subjects.
   a. Write down two classes with the mentioned properties. (all private)
   b. Write necessary public setter functions
   c. Write a non-member compare(…) function that will take a science student object and a commerce student object, and return the roll number of the student having higher average overall marks.
7. Solve the problem using Object Oriented Paradigm.
   Given an integer n, find the value of
   a. $1 + 2 + 3 + \dots\dots\dots + n$
   b. $1^2 + 2^2 + 3^2 + \dots\dots\dots + n^2$
   c. $1^3 + 2^3 + 3^3 + \dots\dots\dots + n^3$
8. Solve the problem using Object Oriented Paradigm.
   A library has some books on Data Structure and some books on Algorithms.
   - Books can be issued (taken from library)
   - Books can be returned (to the library)
   - If book-count falls below 2, a warning is shown
   - If book-count is zero, it cannot be issued from the library
9. Solve the problem using Object Oriented Paradigm.
   A line consists of two points. The length is given by:
   $\sqrt{((x1-x2)^2 + (y1-y2)^2)}$
   The midpoint is given by:
   $((x1 + x2) / 2, (y1 + y2) / 2)$
10. Solve the problem using Object Oriented Paradigm.
    An undergraduate course has 70 classes (14wk x 5d). A student can be present, or absent in any of the days. Devise a way to keep track of the attendance of a student.
    - The student can be marked present or absent on day n
    - The total number of present/absent has to be calculated
    - Check if the student is dis-collegiate (<75%)
    - Check if the student is non-collegiate (<90% & > 75%)
11. Problem based on namespace:
    We all know about the built in function strlen(**strg1**). It determines the length of a particular string **strg1** (with spaces). Now you have to make your own function `int func(`**`strg1`**`)` in a class that will find the length of a string without the spaces.
    <u>Example:</u>
    Input: My name is x.
    Built in function Output: 13
    Own function Output: 10
12. Irteza likes to travel many places. Sometimes for efficient travelling he needs to travel in minimum time or minimum cost. Naturally, he travels by BUS or TRAIN. So, now you have to help him to find out which vehicle satisfies his requirement for efficient travelling.
    a. Write down two classes that represents a BUS and a TRAIN successively (all the attributes will be private)
    b. Each class will have two attributes

        i. Velocity
        ii. Cost per Kilometer

    c. Write a function in each class to set the values.
    d. Write one non-member function time(….) that will calculate the time needed for each vehicle and compare them for finding the efficient vehicle.
    e. Write a non-member function cost(…) that will calculate the cost for each vehicle and compare them for finding the efficient vehicle.

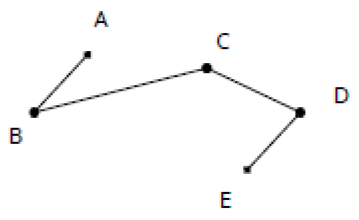
13. In our country it is not possible to transfer money between two different banks. You need to design a system by which transfers can be done between two different banks(Such as DBBL and Trust Bank). For Simplicity, ignore the account number and just think about balance. Transfer can happen in both directions, i.e., from DBBL to Trust or from Trust to DBBL. This should be specified by user input.
    a. Write down two classes that represents 2 different bank accounts such as DBBL and Trust Bank (all the attributes will be private)
    b. Each class will have 1 attribute -Balance
    c. Write functions in each class to set and get the values.
    d. In the main function you need to take the amount of money a user wants to transfer.
    e. Write one non-member function transferMoney(….) that will update the Balance of each bank after each transfer.
    f. Finally, show the current balance of each bank.
14. A straight line consists of two endpoints. However, it may consist of a number of line segments. In this problem, you'll have to work with a line containing several line segments.
    For example, take a look at the whole line AE(the right image below). It consists of four line segments (AB, BC, CD and DE).





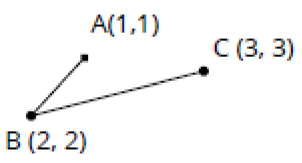
    You'll need to design a class that will hold a line consisting of line segments. There can be n number of line segments. The Constructor will receive the value of n. Then you'll write a friend function of this class that will calculate the total length of this line. The total length of AE is AB+BC+CD+DE.
    As the number of points are arbitrary, you'll need to use dynamic memory to hold the coordinates of the point in line segments (e.g. the coordinates of point A, B, C, D and E in the figure below). The following can be the skeleton of the class.

```
class Line
{
    int segments;
    int *x;
    int *y;
public:
    Line(int n)
    {
        segments = n;
        //initialize x and y
    }
    //Write the destructor and copy constructor here

    //Set the coordinate of i'th point
    void setPoint(int i, int _x, int _y)
    {
        x[i] = _x;
        y[i] = _y;
    }

    friend double getLength(Line l);
};
```



Example

n number of segments will have n+1 endpoints.
The following line AC has 3 points.



Then in the main function, we'll write:

```
Line line1(2);    //because there are 2 segments: AB and BC
line1.setPoint(0,1,1);  //0th point, A: (1,1)
line1.setPoint(1,2,2);  //1st point, B: (2,2)
line1.setPoint(2,3,3);  //2nd point, C: (3,3)
cout << getLength(line1) << endl;   // 2.82843
```

Make sure the following main function works.

```
int main()
{
    int n;
    cout << "Enter the number of segments in the line: ";
    cin >> n;
    Line line1(n);
    for (int i = 0; i <= n; i++)
    {
        int x, y;
        cout << "Enter the x coordinate of "<< (i+1) << "th point: ";
        cin >> x;
        cout << "Enter the y coordinate of "<< (i+1) << "th point: ";
        cin >> y;
        line1.setPoint(i, x, y);
    }
    double length = getLength(line1);
    cout << "The length of the line is " << length << endl << endl;
}
```