Follow Coding Standards

Day 6 Assignments

**16.** Write a program that reads in string input from the user; reverse the case of the letters, and then echoes the string back to the user.

Start by creating a class called in out to handle these tasks. It needs:

 One data member: a character buffer 80 bytes in length (as represented by a const value).

 A function called read() (defined inside the class definition) that uses the C function getchar() to read in the characters. The function should return “true” if end-of file is found, “false”, otherwise

 A function called convert() (defined outside the class definition) that converts all lower case letters to upper case, all upper case letters to lower case, and does not modify any non-letters.

 A function called print() (defined inside the class definition) that outputs the contents of the buffer.

To test your class, use the following main() function:

int main()

{

cout<<"Enter some data"<<endl;

CInout io;

while(!io.read())

{

io.convert();

io.print();

}

return 0;

}

**17**.Write a program that computes number of 25ps, 50ps,1 Re,2 Rs and 5 Rs of coins for a given amount of money. Define a class

class CCoin

{

int denomination;

int count;

};

Create an array of CCoin class objects for each denomination(5,2,1,50ps,25ps)

The function void initialize(int ) initializes the denominator and make count = 0 for each denomination.

For each amount entered, call the function change( float& ) . The function updates the largest count of the given coin for the amount and decrement the amount.

void print() prints the count for each denominations. Do not print for denominations with count 0. Use the main given below.

void main()

{

CCoin carray[5];

//Call initialize function for each object in the array.

cout << “ Enter the amount :“;

cin >> amount;

//Call change function for calculating the denomination.

//Print the denomination and count for each CCoin object.

}

For Eg: if the amount entered is 27.75 the output is as follows :

Denomination Count

5 5

2 1

1 0

0.5 1



**18.** Create a class to hold a point containing member variable to store x and y co-ordinates. Add necessary member functions to initialize, set and to get values.

Create a class to represent rectangle that contains member variables to hold two instances of point (top left and bottom right point).Add necessary member function to handle rectangle instance.

Implement a main() to create as many instances of rectangle as per user’s choice dynamically. Then calculate the perimeter of those rectangles.

Formula is 2(length + breadth).

Then add a global function sort to arrange to display those instances in descending order on the basis of their perimeter.