CS 112 Spring 2022 Semester -- Midterm Examination -- Wednesday 6.4.2022 -- Time: 60 Minutes--- Marks: 30

You are advised to READ these notes:

1. Attempt all the questions on the answer sheet provided.
2. After asked to commence the exam. , please verify that you have one (1) printed page with 5 questions.
3. The invigilator present is not supposed to answer any questions. No one may come to you for corrections and you are not supposed to request to call anyone. Make assumptions wherever required and clearly mark them.

Q1: *[Points: 10, Difficulty level: Medium-Difficult, CLO-1]* In architectural drawings, the distances are measured in feet and inches according to the English system of measurement. There are 12 inches in a foot. The length of a living room, for example, might be given as 15’–8”, meaning 15 feet plus 8 inches. The hyphen is not a negative sign; it merely separates the feet from the inches. Figure 1 shows typical length measurements in the English system. Suppose you want to create a drawing or architectural program that uses the English system, it will be convenient to store distances as two numbers, representing feet and inches. Develop a complete class with proper constructor and destructor functions as well as set and get functions to model and program the Distance representation in the English System. Also include two functions, getDistance to get input from the user and showDistance to display the distance in the prescribed format. The class should also provide the following overloaded operator capabilities:
Overload the addition operator (+) to add two Distances.
Overload the less than operator (<) to compare two Distances
Overload the addition assignment operator (+=)

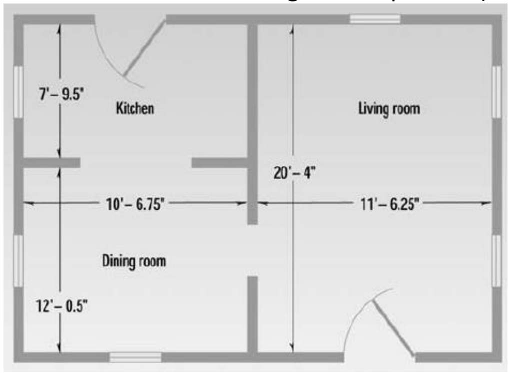

Figure 1: English System of Measurement

Solution:

```
#include <iostream>
using namespace std;
class Distance
{
	private:
		int feet;
		float inch;
	public:
	Distance();
	Distance(int a,float b);
	void setDistance();
	int getFeet();
	float getInch();
	void distanceSum(Distance d);
};
int main()
{
	Distance D1,D2;
	D1.setDistance();
	D2.setDistance();
	D1.distanceSum(D2);
	return 0;
}
/*Function Definitions*/
Distance::Distance()
{
	inch=feet=0;
}
Distance::Distance(int a,float b)
{
	feet=a;
	inch=b;
}
void Distance::setDistance()
{
	cout<<"Enter distance in feet";
	cin>>feet;
	cout<<endl<<"Enter inches:";
	cin>>inch;
}
int Distance::getFeet()
{
	return feet;
}
float Distance::getInch()
{
	return inch;
}
void Distance::distanceSum(Distance d)
{
	cout<<"feet="<<d.feet+feet<<endl;
	cout<<"inches="<<d.inch+inch<<endl;
}
```

Q2: *[Points: 5, Difficulty level: Medium, CLO-2]* Write a function that receives two character pointers having the following prototype.
int compare (char *, char *);

The function receives two words and the task of the function is to compare these two words character-by-character.
If both the words are the same, the function returns 100,
If the first word is larger than the second, the function return 500,
If the second word is larger than the first, the function return 10.
Hint: Among the words “Alpha” and “Gama”, “Gama” is larger.

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

int compare(char *s1, char *s2){
  int i =0, j =0 ;

  while(s1[i] != '\0' && s1[j] != '\0'){
    if(s1[i] > s2[j])
      return 500;
    else if(s1[i] < s2[j])
      return 10;
    ++i;
    ++j;
  }
  if(s1[i] == '\0' && s1[j] == '\0')
    return 100;

  else if(s1[i] != '\0')
```

```
    return 500;

  else
    return 10;

}
int main() {
    cout << compare((char *)"Alpha", (char *)"Gama")<<endl;
    cout << compare((char *)"Alpha", (char *)"Alpha")<<endl;
    cout << compare((char *)"Gama", (char *)"Alpha")<<endl;
}
```

Q3: *[Points: 5, Difficulty level: Medium, CLO-2]* A palindrome is a string that is spelled the same way forward and backward. Some examples of palindromes are "radar," "able was i ere i saw elba" and (if blanks are ignored) "a man a plan a canal panama." Write a recursive function `testPalindrome` that returns `true` if the string stored in the array is a palindrome, and `false` otherwise. The function should ignore spaces and punctuation in the string.

Solution:

```
using namespace std;

// A recursive function that
// check a str[s..e] is
// palindrome or not.
bool isPalRec(char str[],
              int s, int e)
{

    // If there is only one character
    if (s == e)
    return true;

    // If first and last
    // characters do not match
    if (str[s] != str[e])
    return false;

    // If there are more than
    // two characters, check if
    // middle substring is also
    // palindrome or not.
    if (s < e + 1)
    return isPalRec(str, s + 1, e - 1);

    return true;
}

bool isPalindrome(char str[])
{
    int n = strlen(str);

    // An empty string is
    // considered as palindrome
    if (n == 0)
        return true;

    return isPalRec(str, 0, n - 1);
}

// Driver Code
int main()
{
    char str[] = "geeg";

    if (isPalindrome(str))
    cout << "Yes";
    else
    cout << "No";

    return 0;
}
```

Q4: *[Points: 10, Difficulty level: Medium, CLO-1]* Create a class TicTacToe that will enable you to write a complete program to play the game of tic-tac-toe. The class contains as private data a 3-by-3 two-dimensional array of integers. The constructor should initialize the empty board to all zeros. Allow two human players. Wherever the first player makes a moves, place a 1 in the specified square. Place a 2 wherever the second player moves. Each move must be to an empty square. After each move, determine whether the game has been won or is a draw.

Solution:

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

char square[10] = {'o','1','2','3','4','5','6','7','8','9'};

int checkwin();
void board();

int main()
{
          int player = 1,i,choice;

    char mark;
    do
    {
        board();
        player=(player%2)?1:2;

        cout << "Player " << player << ", enter a number:  ";
        cin >> choice;

        mark=(player == 1) ? 'X' : 'O';

        if (choice == 1 && square[1] == '1')

            square[1] = mark;
        else if (choice == 2 && square[2] == '2')

            square[2] = mark;
        else if (choice == 3 && square[3] == '3')

            square[3] = mark;
        else if (choice == 4 && square[4] == '4')

            square[4] = mark;
        else if (choice == 5 && square[5] == '5')

            square[5] = mark;
        else if (choice == 6 && square[6] == '6')

            square[6] = mark;
        else if (choice == 7 && square[7] == '7')

            square[7] = mark;
        else if (choice == 8 && square[8] == '8')

            square[8] = mark;
        else if (choice == 9 && square[9] == '9')

            square[9] = mark;
        else
        {
            cout<<"Invalid move ";
            player--;
            cin.ignore();
            cin.get();
        }
        i=checkwin();

        player++;
    }while(i==-1);
    board();
    if(i==1)

        cout<<"==>\aPlayer "<<--player<<" win ";
    else
        cout<<"==>\aGame draw";

    cin.ignore();
    cin.get();
    return 0;
}

/*********************************************
    FUNCTION TO RETURN GAME STATUS
    1 FOR GAME IS OVER WITH RESULT
    -1 FOR GAME IS IN PROGRESS
    O GAME IS OVER AND NO RESULT
**********************************************/

int checkwin()
{
    if (square[1] == square[2] && square[2] == square[3])

        return 1;
    else if (square[4] == square[5] && square[5] == square[6])

        return 1;
    else if (square[7] == square[8] && square[8] == square[9])

        return 1;
    else if (square[1] == square[4] && square[4] == square[7])

        return 1;
    else if (square[2] == square[5] && square[5] == square[8])

        return 1;
    else if (square[3] == square[6] && square[6] == square[9])

        return 1;
    else if (square[1] == square[5] && square[5] == square[9])

        return 1;
    else if (square[3] == square[5] && square[5] == square[7])

        return 1;
    else if (square[1] != '1' && square[2] != '2' && square[3] != '3'
                    && square[4] != '4' && square[5] != '5' && square[6] != '6'
                  && square[7] != '7' && square[8] != '8' && square[9] != '9')

        return 0;
    else
        return -1;
}


/*******************************************************************
     FUNCTION TO DRAW BOARD OF TIC TAC TOE WITH PLAYERS MARK
********************************************************************/


void board()
{
    system("cls");
    cout << "\n\n\tTic Tac Toe\n\n";
    cout << "Player 1 (X)  -  Player 2 (O)" << endl << endl;
    cout << endl;

    cout << "     |     |     " << endl;
    cout << "  " << square[1] << "  |  " << square[2] << "  |  " << square[3] << endl;

    cout << "_____|_____|_____" << endl;
    cout << "     |     |     " << endl;

    cout << "  " << square[4] << "  |  " << square[5] << "  |  " << square[6] << endl;

    cout << "_____|_____|_____" << endl;
    cout << "     |     |     " << endl;

    cout << "  " << square[7] << "  |  " << square[8] << "  |  " << square[9] << endl;

    cout << "     |     |     " << endl << endl;
}
```

Q5: *Bonus question [Points: 2]* If a plane crashes on the border between the Pakistan and Iran, where do they bury the survivors?

Solution: Survivors are alive, hence will not be buried