

Lecture 17

Project & Activity



QUIZ

قَالَ رَبِّ اشْرَحْ لِي صَدْرِي ۝
﴿٢٥﴾

[فَالَّذِي نَسِيَ كَهُولَ دَعَى رَبَّهُ أَشْرَحَ لَهُ مَنْ يَرَى لِي صَدْرِي مِيرَا سِينَهُ]

وَيَسِّرْ لِي آمْرِي ۝
﴿٢٦﴾

[وَيَسِّرْ لَهُ آمْرِي مِيرَا كَامَ لِي مِيرَا سِينَهُ]

وَاحْلُلْ عُقْدَةً مِنْ لِسَانِي ۝
﴿٢٧﴾

[وَاحْلُلْ لَهُ كَهُولَ دَعَى عُقْدَةً گَرَهَ مِنْ لِسَانِي مِيرَا زِبانَ سِينَهُ]

يَفْقَهُوا قَوْلِي ۝
﴿٢٨﴾

[يَفْقَهُوا وَهُوَ سِجْهَ سَكِينَ [قَوْلِي مِيرَا بَاتَ سِينَهُ]

4 QUESTIONS / FEEDBACK / CONCERNS



INFORMATION
TECHNOLOGY
UNIVERSITY

SE SECA SLIDE OF FAME

5



NO ONE
WEEK - 1



Muhammad Daniyal
Hammad (BSSE23046)
WEEK - 2



Syed Hashim Abbas
(BSSE23084)
WEEK - 3



Umar Ahmad
(BSSE23032)
WEEK - 4



Umar Ahmad
(BSSE23032)
WEEK - 5



Fatima Noorulain
BSSE23003
WEEK - 6



Umar Ahmad
(BSSE23032)
WEEK - 7



YOUR NAME
WEEK - 8



YOUR NAME
WEEK - 9



YOUR NAME
WEEK - 10



YOUR NAME
WEEK - 11



YOUR NAME
WEEK - 12



YOUR NAME
WEEK - 13



YOUR NAME
WEEK - 14



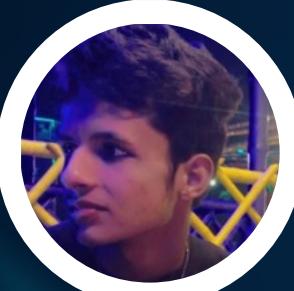
YOUR NAME
WEEK - 15

SE SEC B SLIDE OF FAME

6



Muhammad Mukarram
BSSE23029
WEEK - 1



Muhammad Abdullah
(BSSE23087)
WEEK - 2



Muhammad Abdullah
(BSSE23087)
WEEK - 3



Fasiha Rohail
(BSSE23041)
WEEK - 4



Muhammad Abdullah
(BSSE23087)
WEEK - 5



Hazira Azam
BSSE23019
WEEK - 6



Jamshaid Ahmed
BSSE23012
WEEK - 7



YOUR NAME
WEEK - 8



YOUR NAME
WEEK - 9



YOUR NAME
WEEK - 10



YOUR NAME
WEEK - 11



YOUR NAME
WEEK - 12



YOUR NAME
WEEK - 13



YOUR NAME
WEEK - 14



YOUR NAME
WEEK - 15

RECAP

GitHub

Tools (Cygwin, IDE, GitHub)

Approach towards a word problem

Flowcharts

Flowcharts Advantages & Disadvantages

Algorithms

Pseudocode

Numbers Systems (Decimal, Binary, Octal & Hexadecimal)

Ten's Complement

Twos Complement

main function

Stream in and stream out operators

if else

Functions

Data Types

Arithmetic Operators

Relational Operators

Loops (While, for , do while)

Nested Loops

Switch cases

RECAP

Function Overloading

Scope of variables

Function Prototype and Definition

Default Value in parameters of functions

Parameters by value vs Parameters by Reference

Recursion

Arrays

2D Arrays / Multi Dimensional Arrays



PROJECT DISCUSSION

TASK 1

A company that wants to send data over the Internet has asked you to write a program that will encrypt it so that it may be transmitted more securely. All the data is transmitted as four-digit integers. Your program should read a four-digit integer in main() entered by the user and encrypt it as follows:

1. Replace each digit with the result of adding 7 to the digit and getting the remainder after dividing the new value by 10. Then swap the first digit with the third, and swap the second digit with the fourth. Then display the encrypted integer.
2. After encryption, the program will ask input an encrypted four-digit integer and decrypts it (by reversing the encryption scheme) to form the original number and return the decrypted number.

TASK 2

A bank account charges \$10 per month plus the following check fees for a commercial checking account:

\$.10 each for fewer than 20 checks

\$.08 each for 20-39 checks

\$.06 each for 40-59 checks

\$.04 each for 60 or more checks

The bank also charges an extra \$15 if the balance of the account falls below \$400(before any check fees are applied). Write a program that asks for the beginning balance and the number of checks written. Compute and display the bank's service fees for the month.

Input validation: Do not accept a negative value for the number of checks written. If a negative value is given for the beginning balance, display an urgent message indicating the account is overdrawn.

ARRAY & POINTERS

```
// arrPtr.cpp
// Outputs addresses and values of array elements.
// -----
#include <iostream>
using namespace std;

int arr[4] = { 0, 10, 20, 30 };

int main()
{
    cout << "\nAddress and value of array elements:\n"
        << endl;

    for( int i = 0; i < 4; i++ )
        cout << "Address: " << (void*) (arr+i)      // &arr[i]
            << " Value: " << *(arr+i)           // arr[i]
            << endl;
}

return 0;
}
```

POINTERS

```
float v[6] = { 0.0, 0.1, 0.2, 0.3, 0.4, 0.5 },
    *pv, x;

pv = v + 4;
*pv = 1.4;
pv -= 2;
++pv;

x = *pv++;
x += *pv--;
--pv;
```

POINTERS

```
float v[6] = { 0.0, 0.1, 0.2, 0.3, 0.4, 0.5 },
    *pv, x;

pv = v + 4;          // Let pv point to v[4].
*pv = 1.4;           // Assign 1.4 to v[4].
pv -= 2;             // Reset pv to v[2].
++pv;                // Let pv point to v[3].  
  
x = *pv++;          // Assign v[3] to x and
                      // increment pv.
x += *pv--;          // Increment x by v[4] and let
                      // pv point to v[3] again.
--pv;                // Reset pv to v[2].
```

```
// textPtr.cpp
// Using arrays of char and pointers to char
// -----
#include <iostream>
using namespace std;

int main()
{
    cout << "Demonstrating arrays of char "
        << "and pointers to char.\n"
        << endl;

    char text[] = "Good morning!",
         name[] = "Bill!";
    char *cPtr = "Hello ";           // Let cPtr point
                                    // to "Hello ".
    cout << cPtr << name << '\n'
        << text << endl;

    cout << "The text \" " << text
        << "\" starts at address " << (void*)text
        << endl;

    cout << text + 6      // What happens now?
        << endl;

    cPtr = name;          // Let cPtr point to name, i.e. *cPtr
                        // is equivalent to name[0]
    cout << "This is the " << *cPtr << " of " << cPtr
        << endl;
    *cPtr = 'k';
    cout << "Bill can not " << cPtr << "!\n" << endl;
    return 0;
}
```

POINTERS