Software Development Lab – II [15B17CI271]
Assignment Sheet
<u>Week 2</u>

COURSE OUTCOMES		COGNITIVE LEVELS
C173.1	Write programs in C++ to implement OOPs concepts related to objects, classes, constructor, destructor, and friend function.	Apply Level (Level 3)
C173.2	Write programs in C++ using OOPs concept like encapsulation, inheritance, polymorphism and abstraction.	Apply Level (Level 3)
C173.3	Write programs in C++ using Standard Template Library.	Apply Level (Level 3)
C173.4	Perform exception handling in C++ programs.	Apply Level (Level 3)
C173.5	Write MySQL queries to perform operations like ADD, DELETE, UPDATE, SELECT on relational databases.	Apply Level (Level 3)

Note:

- **Students are advised to submit their solutions to respective lab faculty. The solution file must be named as "rollno_first name_w2.doc" (here w2 represents week2).**

- **Must use the concepts of dynamic memory allocation using pointers and pointer arithmetic while developing solutions to the following questions.**

Q1. WAP Write a program in C++ to dynamically allocate memory for two variables of data type integer and float using new operator. Print the values of these variables and then deallocate the memory using delete operator.

Q2. Assume there are three integer elements in the array A= {10,100,200} and are stored at addresses 0xbfa088b0, 0xbfa088b4, 0xbfa088b8

Note: The address can vary from system to system.

WAP in C++ to perform following operations using pointer arithmetic.

1. Increment a Pointer (++)
   Sample output after incrementing pointer:
   Address of var[0] = 0xbfa088b0
   Value of var[0] = 10
   Address of var[1] = 0xbfa088b4
   Value of var[1] = 100
   Address of var[2] = 0xbfa088b8
   Value of var[2] = 200

2. Decrement a Pointer (--)
   Sample output after decrementing pointer:
   Address of var[3] = 0xbfdb70f8
   Value of var[3] = 200
   Address of var[2] = 0xbfdb70f4
   Value of var[2] = 100
   Address of var[1] = 0xbfdb70f0
   Value of var[1] = 10

Q3. WAP in C++ to store CGPA of n number of students using dynamic memory allocation. Where n is the number of students entered by the user. Display the details of the students.

Q4. What will be the output of following codes? Give a suitable explanation:

a)	b)
| ```
#include <iostream>
using namespace std;
int main()
{
    char lfc [20];
inti;
    for(i = 0; i< 10; i++)
        *(lfc + i) = 65 + i;
    *(lfc + i) = ' ';
cout<<lfc;
    return(0);
}
``` | ```
#include <iostream>
    using namespace std;
int main()
    {
        char *ptr;
        char Str[] = "abcdefg";
ptr = Str;
ptr += 5;
cout<<ptr;
        return 0;
    }
``` |

Q5. WAP to find the largest integer from amongst *n* integers stored in an array.

Q6. WAP to calculate the sum of n integers stored in an array. Ask user to enter the n integers at run time.

Q7. WAP to find out whether a given integer is present in a given array of integers or not.A suitable message should be displayed on screen for both the cases. If the integer is present, then print all the indexes at which it is present in the array. Ask user to input the integer to be searched.[*Use static array with pre-defined integer values*]

Q8. Modify the program in Q3, to read the array dynamically at run time, i.e., ask the user to enter n integer values.

Q9. WAP to sort a given array in ascending order using pointers. Then reverse the sorted array using pointers and print the original, sorted and reversed arrays. [*Use static array with pre-defined integer values*]

Q10. What will be the output of following codes? Give a suitable explanation:

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int arr[] = { 3, 5, 6, 7, 9 };
    int *p = arr;
    int (*ptr)[5] = &arr;

    cout << "p = "<< p <<", ptr = " << ptr << endl;
    cout << "*p = "<< *p <<", *ptr = " << *ptr << endl;

    cout << "sizeof(p) = "<< sizeof(p) <<
            ", sizeof(*p) = " << sizeof(*p) << endl;
    cout << "sizeof(ptr) = "<< sizeof(ptr) <<
        ", sizeof(*ptr) = " << sizeof(*ptr) << endl;
    return 0;
}
```

Q11. WAP to dynamically allocate memory to a 2D array and print the values.

Q12. WAP to print elements of a 2-D array by scripting a pointer to an array.

Hint: `int (*ptr)[4];`

> `ptr = arr;`

Q13. WAP to print the values and address of each element of a 3-D array.