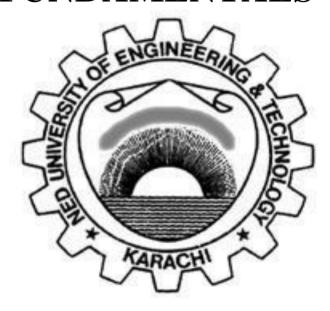
# Practical Workbook

# CT-175 PROGRAMMING FUNDAMENTALS



Name: Hamna Ali Khan

Year: 2024

Batch: 2024

Roll No: <u>CT-157</u>

Department: BCIT

Dept. of Computer Science & Information Technology

NED University of Engineering & Technology

NED University of Engineering & Technology - Department of Computer Science & Information Technology

## EXERCISE Q# 01

- 1. For each of the following, write a single statement that performs the indicated task. Assume that long integer variables value1 and value2 have been defined and that value1 has been initialized to 200000.
- a) Define the variable lPtr to be a pointer to an object of type long.
- b) Assign the address of variable value 1 to pointer variable lPtr.
- c) Print the value of the object pointed to by lPtr.
- d) Assign the value of the object pointed to by lPtr to variable value2.
- e) Print the value of value2.
- f) Print the address of value1.
- g) Print the address stored in lPtr. Is the value printed the same as the address of value 1?

```
#include<stdio.h>
int main(){
long value1=200000,value2; //value1 has been initialized to 200000.
long* lPtr; //Define the variable lPtr to be a pointer to an object of type
long.
lPtr= &value1; //Assign the address of variable value1 to pointer variable lPtr.
printf("The value pointed to by lPtr is : %ld\n", *lPtr); //Print the value of
the object pointed to by 1Ptr.
value2= *lPtr; //Assign the value of the object pointed to by lPtr to variable
value2.
printf("The value pointed to by 1Ptr to value 2 is: %ld\n", value2); //Print the
value of value2.
printf("The address of value1 is: %p\n", (void *)&value1); //Print the address
of value1.
printf("The address stored in lPtr is : %p\n",(void *)lPtr); //Print the address
stored in 1Ptr.
return 0;
```

### **OUTPUT:**

```
The value pointed to by lPtr is: 200000
The value pointed to by 1Ptr to value 2 is: 200000
The address of value1 is: 0061FF14
The address stored in lPtr is: 0061FF14
```

g) The address of value1 and IPtr is same.

NED University of Engineering & Technology - Department of Computer Science & Information Technology

# EXERCISE Q# 02

2. Write a program to implement the exchange or swap the values of 3 variables{a,b,c}. To implement this, you need to write a function called swaped(). void swaped(int \*aPtr, int \*bPtr, int \*cPtr);

```
such that; b ----> temp
a ----> b
c ----> a
temp -> a
#include<stdio.h>
void swapped(int* a,int* b, int* c){
    int temp;
   temp = *b; // Store the value of b in temp
   *b = *a; // Assign value of a to b
    *a = *c;
                   // Assign value of c to a
    *c = temp; // Assign value of temp (original b) to c
}
int main(){
   int a,b,c;
printf("Enter a: ");
scanf("%d", &a);
printf("Enter b: ");
scanf("%d", &b);
printf("Enter c: ");
scanf("%d", &c);
swapped(&a,&b,&c);
printf("After swapping:\n");
printf("a = %d\n", a);
printf("b = %d\n", b);
printf("c = %d\n", c);
return 0;
}
```

### **OUTPUT:**

```
Enter a: 1
Enter b: 2
Enter c: 3
After swapping:
a = 3
b = 1
c = 2
```

NED University of Engineering & Technology – Department of Computer Science & Information Technology

# EXERCISE Q# 03

Write a program that calculates the sum of all the elements in array using pointers.

```
#include <stdio.h>
int Sum(int *arr, int n) {
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += *(arr + i);
    return sum;
}
int main() {
    int n;
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);
    int arr[n];
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int sum = Sum(arr, n);
    printf("The sum of all elements in the array is: %d\n", sum);
    return 0;
}
```

### **OUTPUT:**

```
Enter the number of elements in the array: 4
Enter the elements of the array:
1
2
3
4
The sum of all elements in the array is: 10
```

NED University of Engineering & Technology – Department of Computer Science & Information Technology

# **EXERCISE Q# 04**

Consider the following program:

#include <stdio.h>

int main( void ) {

int \*p1;

char \*p2;

p2=p1;

}

Will the program compile successfully without warnings? Why and why not?

```
#include <stdio.h>
int main( void ) {
int *p1;
char *p2;
p2=p1;
}
```

### **OUTPUT:**

```
q1.c: In function 'main':
q1.c:5:3: warning: assignment from incompatible pointer type [-Wincompatible-pointer-types]
p2=p1;
^
```

This error arises because the pointer 1 and 2 are not of same data type. We have to cast the p1 as char if want desirable output as shown below.

```
#include <stdio.h>
int main( void ) {
int *p1;
char *p2;
p2=(char*)p1;
}
```

It compiles successfully without any warnings now!