**THIS WORK IS FOR INDIVIDUAL STUDENT**

**(Informal Part One of Assignment 3)**

**(Informal Part 1 of Quiz Three will be in Next Class on Tue May 20)**

**DUE, HANDWRITTEN ONLY, IN CLASS Tue May 20, 2025**

**PLEAE ADD THE FOLLOWING TITLE PAGE TO ALL YOUR SUBMISSION OR IT WILL NOT BE GRADED:**

**SPRING 2025 PF SECTION B16**

**TITLE: INFORMAL PART ONE of ASSIGNMENT 3 (IP1A3)**

**DATE: Tue May 20, 2025**

**NAME: Your Full Name**

**REGISTRATION : Your Complete Registration**

**SECTION: B 16**

**CONTENTS: (provide details about what is being submitted):**

**Task One: Number of Question Answered and the number of HOURS spent**

**Task Two: Number of Dry Run Submitted and the number of HOURS spent**

## Task One: CONCEPT TOPICS: C++ Pointers:

Use ChatGPT to generate **Answers (you may be asked to show or print this ChatGPT session)** on C++ Pointers to Understand the following:

1. What is a variable and data type?
2. What is a Pointer Variable?
3. What is **dereferencing** of a Pointer?
4. What is Stack and Heap Computer Memory?
5. What is a Dynamic Variable and
6. What is a Dynamic Array in C++?
7. How to Pass Pointers to Functions?

**Write in your own words a complete understanding of the ChatGPT Answers but without any C++ Code**

## *Task Two: Class Activity Problems*

### Dry Run the following Six codes (must use the same style and details as in the slides as well as in a previously shared document)

### Code One

```
1.  double D = 5.5;
2.  double *pD = & D;
3.  * pD = 14.5;
4.  char word[] = “Ahmad“;
5.  char *pWord = word;
6.  cout << word << “ “ << pWord << endl;
7.  pWord = & word[1];
8.  cout << pWord << endl;
9.  pWord = (word + 4);
10. *pWord = ‘Q’;
11. pWord[1] = ‘r’;
12. cout << pWord << endl;
13. pWord = (word + 6);
14. cout << pWord << endl;
```

### Code Two

```
1.  int main()
2.  {
3.    int values[5] = {1,2,3};
4.    int size = 3;
5.    int * pValues = values;
6.    for(int i = 0; i < size; i++)
7.      {
8.            values[i] = (i - values[i]);
9.      }
10.   for(int i = 0; i < size; i++)
11.     {
12.          pValues[i] = (i - pValues[i]);
13.     }
14. pValues = (values + 1);
15. for(int i = 0; i < size; i++)
16.     {
17.         pValues[i] = i;
18.         cout << pValues[i] << " ";
19.     }
20.   cout << endl;
21.   return 0;
22. }
```

## Code Three

```
1.  int main()
2.  {
3.    char STR[] = "abc";
4.    char * pSTR = STR;
5.    int length = 0;
6.    while(*pSTR != '\0')
7.      {
8.            length++;
9.            pSTR++;
10.     }
11.   cout << length << " " << pSTR << " " << STR << endl;
12.   return 0;
13. }
```

## Code Four

```
1.  int main()
2.  {
3.    int A[5] = {1,-2,3};
4.    int no = 3;
5.    int sum = 0;
6.    int * P = A;
7.    for(int i = 0; i < no; i++)
8.      {
9.          sum += *(P + i);
10.     }
11.   cout << sum << endl;
12. int main()
13. {
14.   int A[5] = {1,-2,3};
15.   int no = 3;
16.   int sum = 0;
17.   int * P = A;
18.   for(int i = 0; i < no; i++)
19.     {
20.         sum += *(P + i);
21.     }
22.   cout << sum << endl;
23.
```

## Code Five: Here identify Errors, Correct these and Continue the Dry Run

```
1.  int A = 10;
2.  double B = 5.5;
3.  int *P;
4.  P = &B
```

```
5.  double * Q = B;
6.  int * R = & A;
7.  R++;
8.  double S = & B;
9.  *S = 12.5;
10. S[1] = 12.5;
11. *(R + 1) = 5;
12. R[0] = 24;
```

## Code Six

```
1.  int main()
2.  {
3.    char A[] = "abc";
4.    char B[5] = "";
5.    char * P = A;
6.    char * Q = B;
7.   while(*P != '\0')
8.   {
9.          *Q = *P;
1.  if(*Q >= 'a' && *Q <= ‘z’)
2.         {
3.                    *Q -= 32;
4.               }
5.          cout << *P << " " << *Q << endl;
6.          Q++;
7.          P++;
8.      }
9.    cout << *P << " " << *Q << endl;
10.  cout << A << " " << B << endl;
11.  return 0;
12. }
```