Adolfo Sanpedro Cruz

Palomar College
Computer Science &
Information Technology
CSCI 112 Programming Fundamentals I
Chapter 6 – HW 8

1. Trace the execution of the following fragment.
   Draw a representation of memory to visualize what the values in memory are including pointer variables and where they point to.

```
int m = 10, n = 5;
int *mp, *np;
mp = &m;
np = &n;
*mp = *mp + *np;
*np = *mp - *np;
printf("%d %d\n%d %d\n", m, *mp, n, *np);
```

```
#define _CRT_SECURE_NO
#include <stdio.h>
```

int m = 10 4 15 4
int n = 5 4
int *mp 10 4
int *np

*mp = *mp + *np
*np = *mp - *np

print(" ", m, *mp, n, *np)

~~5 10 15 5 5~~

1015 15 10 10

Palomar College
Computer Science &
Information Technology
CSCI 112 Programming Fundamentals I
Chapter 6 – HW 8

2. Write the main, function call, and function sumNavg that has three type double input parameters (n1, n2, n3) and two output parameters (n4, n5). The function computes the sum and the average of its three input arguments and relays its results through two output parameters. In essence, input parameters are pass by value (no pointer), and output parameters are pass by reference (pointers).

```
_CRT_SECURE_NO_WARNINGS
<stdio.h>

int main()
{

    double n1 =0;
           n2=0;
           n3=0;
           n4=10;
           n5=20;
  Printf(enter 3 nums);
  scanf("%lf %lf %lf", &n1, &n2, &n3)
  SumNavg(n1, n2, n3, &n4, &n5);
}
void sumNavg(double n1, n2, n3, *n4, *n5)
{
   double sumAll = 0.0; avg=0.0;
    sumAll = n1+n2+n3
    avg = sumAll/3.0

    *n4 = sumAll;
    *n5 = avg;
   Printf("values are %lf %lf %lf %lf %lf", n1,n2,n3,
                                            *n4, *n5)
}
```

Palomar College
Computer Science &
Information Technology
CSCI 112 Programming Fundamentals I
Chapter 6 – HW 8

3. Write the main, function call, function that computes the average of three input number parameters (n1, n2, n3) and has one output parameter sum. Sum calculates the sum of the three numbers. The name of the function is calcSum.

```
cnt_ sum, no, warning
<stdio.h>
void SumCalc (int n1, n2, n3, *sum);

int main()
{ int n1=0, n2, =0, n3=0; int sum=0;
  print (enter 3 num);
  scanf ( %d %d %d, n1, n2, n3);
}  SumCalc ( n1, n2, n3, &sum);

void SumCalc (int n1, n2, n3, *sum)
{       double avg = 0.0;
        *sum = n1+n2+n3;
}       avg = *sum / 3.0;
```

Palomar College
Computer Science &
Information Technology
CSCI 112 Programming Fundamentals I
Chapter 6 – HW 8

4. Write a main, function call, and function called getInput. Create a function that passes two output parameters n1, n2. The getInput function calls scanf to ask the user for input. If both values are greater than 0 the function returns a 1, otherwise 0. Remember, scanf is different syntax when they are output or pass by reference parameters. (hint: no & in front of the variable, as it is already an address)

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>

int main()
{
    int n1=0, n2=0, status=0;
    status = getInput(&n1, &n2);
    printf("%d", status);
    return(0);
}
int getInput(int *n1, int *n2)
{
    printf("Please enter 2 nums");
    scanf("%d %d", n1, n2
    if(*n1>0 && *n2>0)
        return(1);
    else
        return(0);
}
```