REDUCING COMPLEXITY PRINCIPLES

MODULARITY

PROGRAMMING

Write your own program using at least 3 different functions that perform distinctive tasks. Here is a short example:

```
int Add( int x, int y)
£
         int s;
         s= x + y;
        return s;
}
int Mult(int x, int y)
ł
         int s;
         s= x*y;
         return s;
float Av(int x, int y)
£
         int sum;
         float avg;
         sum = Add(x,y);
        avg = sum /2.0;
        return avg;
}
int Big(int x,int y)
{
         if (x \ge y)
                 return x;
         else
                 return y;
int main()
{
         int n1, n2, total,result;
         float ans;
        n1=5;
        n2=6;
         total = Add(n1,n2);
        printf("Sum is %d\n", total);
        result=Mult(n1,n2);
printf("Product is %d\n",result);
        ans= Av(n1,n2);
printf("Average is %f\n",ans);
        result= Big(n1, n2);
         printf("Larger value is %d\n", result);
return 0;
}
```



WHAT TO SUBMIT

Submit the program using notepad.

Community Colleges (C5).

This document is licensed with a <u>Creative Commons Attribution 4.0 International License</u> ©2017 <u>Catalyzing Computing and Cybersecurity in</u>