Tyler Plihcik
EE222 HW4

1.

```
//
//
//  EE222HW4
//
//  Created by Tyler Plihcik on 2/20/20.
//  Copyright © 2020 Tyler Plihcik. All rights reserved.
//

#include <stdio.h>


//function prototype
double degreeToRadians(int degree) ;


int main(  )
{

    //declare variables
    double degree ;
    double result ;


    // get degree input from user
    printf( "Enter Degree: " ) ;
    scanf( "%lf" , &degree ) ;

    //check to make sure user input is within acceptable range
    if( degree >= -180 && degree <= 180 )
    {
        //call calculation function
        result = degreeToRadians( degree ) ;

        //print the result
        printf( "%lf degrees is equal to %lf radians\n\n " , degree ,
result ) ;
    }

    //if the user input was valid..
    else
    {
        //repromt them for a degree in the specified range
        printf( "Please Enter a value between -180 and 180: " ) ;
        scanf( "%lf" , &degree ) ;

        //calculate the result
        result = degreeToRadians( degree ) ;

        //print the result
        printf( "%lf degrees is equal to %lf radians\n\n " , degree ,
result ) ;
```

```
    }


    return 0;
}


/*
 Funciton: degreesToRadians
 Algorithm: coverts a degree value to a radian value
 */
double degreeToRadians(int degree)
{
    //declare vars
    double pi = 3.14 ;

    double radians = 0.0 ;

    //degree to radian conversion
    radians = degree * ( pi / 180 ) ;

    //return converted value
    return radians ;
}
```

```
Enter Degree: 120
120.000000 degrees is equal to 2.093333 radians

 Program ended with exit code: 0
```

```
Enter Degree: 900
Please Enter a value between -180 and 180: 45
45.000000 degrees is equal to 0.785000 radians

 Program ended with exit code: 0
```

```
Enter Degree: -900
Please Enter a value between -180 and 180: 90
90.000000 degrees is equal to 1.570000 radians

 Program ended with exit code: 0
```

```c
//
//  HW4_Pt2.c
//  EE222HW4
//
//  Created by Tyler Plihcik on 2/20/20.
//  Copyright © 2020 Tyler Plihcik. All rights reserved.
//

#include <stdio.h>


//funciton prototypes
double addition( int numOne , int numTwo ) ;

double subtraction( int numOne , int numTwo ) ;

double multiplication( int numOne , int numTwo ) ;

double division( int numOne , int numTwo ) ;




int main(  )
{

    //constants for switch statement
    const char ADD  = '+' ;
    const char MULTIPLY = '*' ;
    const char DIVIDE = '/' ;
    const char SUBTRACT =  '-' ;


    //declare variables
    char opperand ;
```

```
    int numOne ;
    int numTwo ;

    double result = 0 ;

    printf( "Enter an operator: \n" ) ;
    scanf( "%c" , &opperand ) ;

    printf( "Enter the first number: \n" ) ;
    scanf( "%d" , &numOne ) ;

    printf( "Enter the second number: \n" ) ;
    scanf( "%d" , &numTwo ) ;



    switch( opperand )
    {
        case ADD:
            result = addition( numOne , numTwo ) ;
        break ;

        case SUBTRACT:
            result = subtraction( numOne , numTwo ) ;
        break ;

        case MULTIPLY:
            result = multiplication( numOne , numTwo ) ;
        break ;

        case DIVIDE:
            result = division( numOne , numTwo ) ;
        break ;
    }

    printf( "The result of %d %c %d is: %f\n\n" , numOne ,
opperand , numTwo , result ) ;
}



/*
 Function: addition
 Algorithm: takes in two integers as parameters and adds them
```

```
 */
double addition( int numOne , int numTwo )
{

    return numOne + numTwo ;
}

/*
Function: subtraction
Algorithm: takes in two integers as parameters and subtracts
them
*/
double subtraction( int numOne , int numTwo )
{
    return numOne - numTwo ;
}

/*
Function: multiplication
Algorithm: takes in two integers as parameters and mulitplies
them
*/
double multiplication( int numOne , int numTwo )
{
    return numOne * numTwo ;
}

/*
Function: division
Algorithm: takes in two integers as parameters and divides them

*/
double division( int numOne , int numTwo )
{
    return numOne / numTwo ;
}
```

```
Enter an operator:
/
Enter the first number:
8
Enter the second number:
2
The result of 8 / 2 is: 4.000000

Program ended with exit code: 0
```

```
Enter an operator:
-
Enter the first number:
40
Enter the second number:
2
The result of 40 - 2 is: 38.000000

Program ended with exit code: 0
```

```
Enter an operator:
+
Enter the first number:
4
Enter the second number:
8
The result of 4 + 8 is: 12.000000

Program ended with exit code: 0
```

```
Enter an operator:
*
Enter the first number:
1
Enter the second number:
9
The result of 1 * 9 is: 9.000000

Program ended with exit code: 0
```