Cavendish University Uganda.

Cat 1 October 2024

Course Unit: Principles of C Programming

Name : Lakwenyero Daniel

Student's ID: 203603

Question

Write a program using C programming for a calculator that performs basic operations (addition, subtraction, multiplication, division)

Task:

a) Implement a simple Calculator class.

This is the calculator in C

```
#include <stdio.h>
#include <assert.h>
#include <math.h>

// calculator functions
double add(double a, double b){
    return a+b;
}
double subtract(double a, double b){
    return a-b;
}
double multiply(double a, double b){
    return a*b;
}
double divide(double a, double b){
    if(b==0){
        printf("Error: Division by zero is invalid.\n");
        return 0;}
```

```
    else {
        return a / b; }
    }
// The main part of c calculater.

int main() {
    double a, b, result;
    char operator;
// calculator functionality

printf("Enter an expression (for example 4+5): ");
scanf("%lf %c %lf", &a, &operator, &b);

switch (operator) {

case '+':
    result = add(a, b);
    break;
case '-':
    result = subtract(a, b);
    break;
case '/':
    result = divide(a, b);
    break;
default:
    printf("Invalid operator, Please try again!.\n");
    return 1; }
  printf("Result: %.2f\n", result);
  return 0; }
```

b) Write unit tests for the methods in the calculator.

Here I will add a function that will compare floating-point number with a small epsilon.

```
int is_close(double x, double y, double epsilon){
    return fabs(x-y) < epsilon; }
```

Now below is where I carry out the unit tests for each method.

```
// assert is used in this case
// code.
    void testing_calc(){
    // test addition.
      assert (is_close(add(3,2), 5, 0.0001));
      assert (is_close(add(1.5, 2.5), 4, 0.0001));
    // testing subtraction.
      assert (is_close(subtract(3,2), 1, 0.0001));
      assert (is_close(subtract(2,5), -3, 0.0001));
    // testing multiplication
      assert (is_close(multiply(3,2), 6, 0.0001));
      assert (is_close(multiply(1.5,2), 3, 0.0001));
    // testing division method.
      assert (is_close(divide(10,2), 5, 0.0001));
      assert (is_close(divide(5,2), 2.5, 0.0001));
    // testing division by zero
      assert (divide(5,0) == 0); // error expected here
      Printf("All tests passed successfully! \n");
```

When the tests are all successful, the program will display "All tests passed successfully!", however if it does not successfully pass the tests, then it will display an error.