

March 27, 2009

- Learning objectives. At the completion of this lab, you should be able to
 - Describe the concept of 'Black Box' testing.
 - Recognize the difficulties brought about by poor software design.
 - Describe some benefits of Extreme Programming
- Assignment
 - Divide up into groups of two students and one computer.
 - Download and run SoftEngr_partA.exe
 - You have **10 minutes** to test the program for accuracy and list all errors found:
 - 1_
 - 2_
 - 3_
 - 4_
 - 5_
 - 6_
 - You have **15 minutes** to download SoftEngr_partA.c and repair all computation errors.
 - You have **15 minutes** to add functionality to the program so that the user can also input data for a circle and get back the area and perimeter of the circle.
 - Demonstrate the program to the instructor in class.
 - Download SoftEngr_partB.c
 - Compare the design of the SoftEngr_partA to that of SoftEngr_partB.
 - Class discussion for 10 minutes.
 - Write a brief reflection on the differences in the design of the two programs. Submit your reflection via Blackboard.
 - What obvious design differences did you notice between the two programs?
 - Which program was easier to understand? Why?
 - How do you think 'goto' statements affect the maintainability of programs?
 - Do you think that number of lines of code is a good indicator of program complexity? Explain.
 - What did you learn from that exercise that will change the way you write code in the future?

```
// Tim Hardin
// CECS 230
// Software Engr - Part A
```

```
#include<stdlib.h>
#include<stdio.h>
```

```
int a,b,c,d,e,f;
```

```
int main()
```

```
{
```

```
a=b=c=d=e=f=0;
```

```
a1: printf("Here is the menu\n"); printf(" 1 - square\n");
```

```
a2: printf(" 2 - rectangle\n"); printf(" 9 - quit\n");
```

```
printf("Your choice: "); int r;
```

```
scanf("%d", &r); if ( r == 1 ) goto x4; else goto q1;
```

```
q1: if ( r == 2 ) goto hell0; else goto q2;
```

```
q3: if (( r!=1 ) || ( r!= 2) || ( r!=9 )) goto O4;
```

```
q2: if ( r == 9 ) goto by; else goto q3;
```

```
x4: printf("Enter side of square: ");
```

```
scanf("%d", &a); goto ps;
```

```
hell0: printf("Enter side #1: "); scanf("%d", &b); c=b;
```

```
printf("Enter side #2: "); scanf("%d", &f); a=f; goto rp;
```

```
by: system("pause"); return 0; O4: printf("Try again stupid!!!!\n\n");
```

```
goto a1; rp: printf("Rectangle: ");
```

```
printf(" %d x", a); printf(" %d, perimeter = ", f);
```

```
printf(" %d, area =", 2*b+f*2); f = b*f;
```

```
printf(" %d\n", f); goto a1; ps: b = 2*a+a; c = b/3*a;
```

```
printf("Square: Side = %d, perimeter = %d, area = %d\n\n", a,b,c);
```

```
goto a1;
```

```
system("pause");
```

```
return 0;
```

```
}
```

variable names have no meaning

the dreaded 'goto'

multiple statements
on one line

no indentation

Not to mention a few computation errors!

```

// Tim Hardin
// CECS 230
// Software Engr - Part B

#include<stdio.h>
#include<stdlib.h>

int menu() // display menu and get user response
{
    printf("Here is the menu\n");
    printf(" 1 - square\n");
    printf(" 2 - rectangle\n");
    printf(" 9 - quit\n");
    printf("Your choice: ");
    int response;
    scanf("%d", &response);
    return response;
}

void square() // calc and display info for a square
{
    int side;
    printf("Enter side of square: ");
    scanf("%d", &side);
    int area, perim;
    area = side*side;
    perim = 4 * side;
    printf("Square: Side = %d, perimeter = %d, area = %d\n\n", side, perim, area);
    return;
}

void rectangle() // calc and display info for a rectangle
{
    int sidel, side2;
    printf("Enter side #1: ");
    scanf("%d", &sidel);
    printf("Enter side #2: ");
    scanf("%d", &side2);
    int area = sidel * side2;
    int perim = 2 * ( sidel + side2 );
    printf("Rectangle: %d x %d, perimeter = %d, area = %d\n\n", sidel, side2, perim,
area);
}

int main()
{
    int choice;
    do
    {
        choice = menu();
        if (choice == 9) break;
        else if (choice == 1) square();
        else if (choice == 2 ) rectangle();
        else printf("Bad choice you idiot... try again\n\n");
    } while(1);
    system("pause");
    return 0;
}

```

