Refactoring 1 - Project 1

```
int function(value_A, value_B) {
        int total = 0;
        total = (value_A*value_B);
        total = total/2;
        return total;
}
#define MACRO
void main(void) {
        int value;
        value = function(2
        #ifdef MACRO
                , 10
        #else
                , 20
        #endif
        );
        printf("%d", value);
}
```

Undisciplined

```
int function(value_A, value_B) {
        int total = 0;
        total = (value_A*value_B);
        total = total/2;
        return total;
}
#define MACRO
void main(void) {
        int value;
        #ifdef MACRO
                value = function(2, 10);
        #else
                value = function(2, 20);
        #endif
        printf("%d", value);
}
```

Disciplined

Refactoring 1 - Project 2

```
int function(value_A, value_B) {
        int result = 0;
        result = (value_B-value_A);
        result = result + 20;
        return result;
}
#define MACRO
void main(void) {
        int output;
        output = function(3
        #ifdef MACRO
                , 15
        #else
                , 11
        #endif
        );
        printf("%d", output);
}
```

Undisciplined

```
int function(value_A, value_B) {
        int result = 0;
        result = (value_B-value_A);
        result = result + 20;
        return result;
}
#define MACRO
void main(void) {
        int output;
        #ifdef MACRO
                output = function(3, 15);
        #else
                output = function(3, 11);
        #endif
        printf("%d", output);
}
```

Disciplined

Refactoring 1 - Project 3

```
int function(value_A, value_B) {
        int result = 0;
        result = (value_A-value_B);
        result = result*2;
        return result;
}
#define MACRO
void main(void) {
        int output;
        output = function(6
        #ifdef MACRO
                , 3
        #else
                , 2
        #endif
        );
        printf("%d", output);
}
```

Undisciplined

```
int function(value_A, value_B) {
        int result = 0;
        result = (value_A-value_B);
        result = result*2;
        return result;
}
#define MACRO
void main(void) {
        int output;
        #ifdef MACRO
                output = function(6, 3);
        #else
                output = function(6, 2);
        #endif
        printf("%d", output);
}
```

Disciplined

Refactoring 1 - Project 4

```
int function(value_A, value_B) {
        int total = 0;
        total = (value_A*value_B);
        total = total/2;
        return total;
}
#define MACRO
void main(void) {
        int result;
        result = function(3
        #ifdef MACRO
                , 6
        #else
                , 8
        #endif
        );
        printf("%d", result);
}
```

Undisciplined

```
int function(value_A, value_B) {
        int total = 0;
        total = (value_A*value_B);
        total = total/2;
        return total;
}
#define MACRO
void main(void) {
        int result;
        #ifdef MACRO
                result = function(3, 6);
        #else
                result = function(3, 8);
        #endif
        printf("%d", result);
}
```

Disciplined

Refactoring 2 - Project 1

```
#define MACRO
void main(void) {
        int status, total = 0;
        int extra = 0;
        total = 10;
        if(total > 9
        #ifdef MACRO
                && extra == 1
        #endif
        )
                status = 1;
        else
                status = 0;
        printf("%d", status);
}
```

Undisciplined

```
#define MACRO
void main(void) {
        int status, total = 0;
        int extra = 0;
        total = 10;
        int test = (total > 9);
        #ifdef MACRO
                test = (test == 1) && (extra != 0);
        #endif
        if(test == 1)
                status = 1;
        else
                status = 0;
        printf("%d", status);
}
```

Disciplined

Refactoring 2 - Project 2

```
#define MACRO
void main(void) {
        int value, result = 0;
        int minimum = 1;
        result = (3 * 4)/2;
        if(result < 10
        #ifdef MACRO
                && minimum == 2
        #endif
        )
                value = 0;
        else
                value = 1;
        printf("%d", value);
}
```

Undisciplined

```
#define MACRO
void main(void) {
        int value, result = 0;
        int minimum = 0;
        result = (3 * 4)/2;
        int test = (result < 10);
        #ifdef MACRO
                test = (test == 1) && (minimum == 2);
        #endif
        if(test == 1)
                value = 0;
        else
                value = 1;
        printf("%d", value);
}
```

Disciplined

Refactoring 2 - Project 3

```
#define MACRO
void main(void) {
        int status, total = 0;
        int extra = 0;
        total = 12;
        if(total > 0
        #ifdef MACRO
                && extra == 1
        #endif
        )
                status = 0;
        else
                status = 1;
        printf("%d", status);
}
```

Undisciplined

```
#define MACRO
void main(void) {
        int status, total = 0;
        int extra = 0;
        total = 12;
        int test = (total > 0);
        #ifdef MACRO
                test = (test == 1) && (extra == 1);
        #endif
        if(test == 1)
                status = 0;
        else
                status = 1;
        printf("%d",  status);
}
```

Disciplined

Refactoring 2 - Project 4

```
#define MACRO
void main(void) {
        int value, result = 0;
        int minimum = 0;
        result = (4*4)/2;
        if(result < 10
        #ifdef MACRO
                && minimum == 1
        #endif
        )
                value = 0;
        else
                value = 3;
        printf("%d", value);
}
```

Undisciplined

```
#define MACRO
void main(void) {
        int value, result = 0;
        int minimum = 0;
        result = (4*4)/2;
        int test = (result < 10);
        #ifdef MACRO
                test = (test == 1) && (minimum == 1);
        #endif
        if(test == 1)
                value = 0;
        else
                value = 3;
        printf("%d",  value);
}
```

Disciplined

Refactoring 3 - Project 1

```
void main(void) {
        int output;
        int value_A = 0;
        int value_B = 0;
        #ifdef MACRO
                if(value_A == 1
        #else
                if(value_B == 0
        #endif
                output = 1;
        else
                output = 0;
        printf("%d", output);
}
```

Undisciplined

```
void main(void) {
        int output;
        int value_A = 0;
        int value_B = 0;
        int test;
        #ifdef MACRO
                test = (value_A == 1);
        #else
                test = (value_B == 0);
        #endif
        if(test == 1)
                output = 1;
        else
                output = 0;
        printf("%d", output);
}
```

Disciplined

Refactoring 3 - Project 2

```
void main(void) {
        int status;
        int value_A = 0;
        int value_B = 0;
        #ifdef MACRO
                if(value_A != 0)
        #else
                if(value_B != 1)
        #endif
                status = 0;
        else
                status = 1;
        printf("%d", status);
}
```

Undisciplined

```
oid main(void) {
        int status;
        int value_A = 0;
        int value_B = 0;
        int test;
        #ifdef MACRO
                test = (value_A != 0);
        #else
                test = (value_B != 1);
        #endif
        if(test == 1)
                status = 0;
        else
                status = 1;
        printf("%d", status);
```

Disciplined

Refactoring 3 - Project 3

```
void main(void) {
        int output;
        int value_A = 0;
        int value_B = 10;
        #ifdef MACRO
                if(value_A <
        #else
                if(value_B >
        #endif
                output = 2;
        else
                output = 0;
        printf("%d",  output)
}
```

Undisciplined

```
void main(void) {
        int output;
        int value_A = 0;
        int value_B = 10;
        int test;
        #ifdef MACRO
                test = (value_A < 0);
        #else
                test = (value_B > 0);
        #endif
        if(test == 1)
                output = 2;
        else
                output = 0;
        printf("%d",  output);
}
```

Disciplined

Refactoring 3 - Project 4

```
void main(void) {
        int status;
        int value_A = 0;
        int value_B = 20;
        #ifdef MACRO
                if(value_A > 0)
        #else
                if(value_B > 10
        #endif
                status = 0;
        else
                status = 1;
        printf("%d",  status);
}
```

Undisciplined

```
void main(void) {
        int status;
        int value_A = 0;
        int value_B = 20;
        int test;
        #ifdef MACRO
                test = (value_A > 0);
        #else
                test = (value_B > 10);
        #endif
        if(test == 1)
                status = 0;
        else
                status = 1;
        printf("%d",  status);
}
```

Disciplined