Achieve Target 1

The following is the rewritten version of the badminton training class requirement program.

```c
#include <stdio.h>

void main() {
    int age;
    printf("Enter your age: ");
    scanf("%d", &age);

    if (age < 18 || age >= 65) {
        printf("Sorry. Your age does not satisfy the requirement\n");
    } else {
        printf("Your age satisfies the requirement\n");
    }

    fflush(stdin);
    getchar();
}
```

The solution involves you to think of the opposite condition. If the required age is greater than or equal to 18 and less than 65, then the age not satisfying the requirement is less than 18 or greater or equal to 65.

This is the logic thinking you would use everyday. If you find it confusing, I would suggest you to practice everyday. Whenever you see the rules/condition for something (the condition satisfying half fare or certain benefits/schemes), try to work out the reverse. We all have this ability in logical thinking, but you would need to sharpen it by more practice.
The following is a solution of the bus fare program.

```c
#include <stdio.h>

void main() {

int age;
float fare;

printf("Enter your age: ");
scanf("%d", &age);
printf("Enter the bus fare: ");
scanf("%f", &fare);

if (age < 12 || age >= 65)
    fare = fare / 2; /* pay half fare */
else
    fare = fare * 0.9; /* a 10% discount */

printf("The fare to tender is $%f\n", fare);

fflush(stdin);
getchar();
}
```

 Achieve Target 2

We simply count the number of operators involved in the four if structures in the first program. We would get 8 (1 for the first if structure, 3 for the second and third, and 1 for the last).

The use of if-else in the re-written program changes the count significantly. The minimal number of operators executed is 1. If the value of variable bmi is less than 19, then only the first if statement is executed and the rest would be ignored. The maximum number of operators executed is 7, in the cases when the variable bmi is greater than 25.

The observation is that in generally if-else structure can reduce the execution of a program, therefore making a program faster with the less amount of execution.

Please note that the above numbers are not absolutely correct in reality. The reason will be explained later in the course. Anyway, we are interested in their relative counts only.

The following program is a solution for Promotion Scheme 1. Note that the commented part shows an alternative method of not using if-else structure.

```c
#include <stdio.h>

void main() {

int numberOfCan;
int freeCan;
```
The following program is a solution for Promotion Scheme 2. Again note that the commented part shows an alternative method of not using if-else structure.

```c
#include <stdio.h>

void main() {

    int numberOfCan;
    int freeCan;

    printf("Enter the number of cans purchased: ");
    scanf("%d", &numberOfCan);

    freeCan = numberOfCan / 8;
    if (numberOfCan >= 30)
        freeCan = freeCan + 4;
    if (numberOfCan > 50)
        freeCan = freeCan + 4;

    /*
    freeCan = (numberOfCan / 8) +
              (numberOfCan >= 30) * 4 +
              (numberOfCan > 50) * 4;
    */
    printf("The customer should receive extra %d cans free\n", freeCan);
}
```
The following program is a solution for Promotion Scheme 3. Note that there is the second if structure there to check for the condition where `freeCan` is greater than 10.

```c
#include <stdio.h>

void main() {
    int numberOfCan;
    int freeCan;

    printf("Enter the number of cans purchased: ");
    scanf("%d", &numberOfCan);

    if (numberOfCan > 20 && numberOfCan <= 50) {
        freeCan = (numberOfCan - 20) / 5;
    }
    else if (numberOfCan > 50) {
        freeCan = 6 + (numberOfCan - 50) * 2 / 5;
    }
    if (freeCan > 10)
        freeCan = 10;

    printf("The customer should receive extra %d cans free\n", freeCan);

    fflush(stdin);
    getchar();
}
```

Achieve Target 3

A solution for the rewritten of the Taxi Fare Calculator program using a flat if-else structure. Only the processing part is shown.

```c
...  
    if ((area == 'N') && (distance < 2.0))
        fare = 12.5;
    else if ((area == 'N') && (distance >= 2.0))
        fare = 12.5 + (distance - 2.0) * 5.0;
    else if ((area == 'L') && (distance < 2.5))
        fare = 12.5;
    else if ((area == 'L') && (distance >= 2.5))
        fare = 12.5 + (distance - 2.5) * 4.0;
    else if ((area == 'H' || area == 'K') && (distance < 2.0))
        fare = 15;
    else if ((area == 'H' || area == 'K') && (distance >= 2.0))
```
fare = 15 + (distance - 2.0) * 7.0;
...

Achieve Target 4

Changing the value of variable number from 1 to 2 is a solution.

...  
int number;

    number = 2;
    switch (number) {
        ...

Achieve Target 6

A solution for Prime Number Checker program is given below.

#include <stdio.h>

void main() {

    int number;

    printf("Enter an integer (1 to 10): ");
    scanf("%d", &number);

    switch (number) {
    case 1:
        printf("The number is special\n");
        break;
    case 2: case 3: case 5: case 7:
        printf("The number is a prime number\n");
        break;
    case 4: case 6: case 8: case 9: case 10:
        printf("The number is not a prime number\n");
        break;
    default:
        printf("Entered is not an integer between 1 to 10\n");
    }

    fflush(stdin);
    getchar();
}