

- Previous hand-outs can be downloaded from <http://www.cs.ucr.edu/~ycao/cs10/>.

1 Sentinel-Controlled Loop

Usually, you use a counter to let the loop to iterate for n times. For example:

```
int loop_count = 0;
const int MAX_LOOP = 10;
while (loop_count < MAX_LOOP) {
    // do my work
    loop_count ++;
}
```

However, sometimes, you do not know the value for `MAX_LOOP`. In other words, you do not know for how many iterations a loop should run. For example, your program needs to process grades of several students, and asks the user to input the grades one by one, but you do not know how many students there are. Even if you know there are 51 students, you do not like the idea that your program must be changed when a student dropped the course or a new student enrolls. Using a **sentinel**, you can let the user to determine when the loop should stop, instead of looping for a constant number of times.

A sentinel is a special value, such that when the program gets such a value as input, it should stop the loop. A typical use case would be:

```
// set the sentinel
int const SENTINEL = -1;
// get the input
cin >> input;
// test the input against some sentinel
while (input != SENTINEL) {
    // do my work
    cin >> input;
}
```

After this, as long as the user input a valid value, that is, a value not equal to the sentinel, the loop will continue. The user can input the sentinel value, -1 in this case, to stop the loop.

2 Simple Use of Function

A **function** is a portion of code within a program, which performs a specific task and is relatively independent of the remaining code.

When do we use function? A function is used when there is a independent subtask that is repeated in the program. For example, consider a task of drawing spaceship. This task may repeat many times after you clear the screen. Instead of repeating the same code for many times by dirty copy and paste, you may define a function doing purely that job, and simply call that function when you need to draw the spaceship.

Why do we use function? By introducing functions, you maximize code reuse; by reusing code, you make it easier to make changes, coz now there is only one place to change. Consider you have a mistake in the spaceship. If you draw the spaceship in a function, you only need to update that function; but if you use dirty copy and paste, you have to change every single repeat of that drawing code. Also, by introducing functions, you make each function smaller and more specific. Now the draw function only need to take care of drawing the ship. No getting user input. No initialization of coordiate. No calculation of how to move the ship. Just draw it. Simple and much smaller chance of errors.

Why not loops? First, sometimes you cannot use loops. You will experience this in part 2 of the lab. Second, loop requires you to repeat the task continuously. If you want to do some task once at the beginning of the program, and once at the end, loop won't help. Third, loop does not help when you need to reuse the code among programs. Say you write some very nice code to build a fancy interface to display some error messages in your programming assignment 1. In another programming assignment, you might want to use the same fancy error message. A loop won't help, but a function can be defined and used in many programs.