

CSI 333 – Programming at the Hardware-Software Interface

Storage Allocation/Deallocation in C

Handout 5.1

The segment of code shown below indicates how the standard library functions `malloc` and `free` are used to dynamically allocate and deallocate memory in C programs.

Code segment in C:

```
int *x;
x = (int *) malloc(sizeof(int));
if (x == NULL) {
    printf("Allocation failed.\n"); exit(1);
}
/* Can use *x here. */
.
.
free(x); /* Frees the storage pointed to by x. */
```

Note: A common practice in C is to combine the call to `malloc` and the test of the return value for `NULL` into a single statement as follows.

```
int *x;
if ((x = (int *) malloc(sizeof(int))) == NULL) {
    printf("Allocation failed.\n"); exit(1);
}
/* Can use *x here. */
.
.
free(x); /* Frees the storage pointed to by x. */
```