

```

#include "primitives.h"
#include "object.h"

static INSTANCE_HEADER *instances = NULL;

INSTANCE_HEADER *CreateInstanceHeader (entree, instance_id, width, height)
    ENTRY *entree;
    int instance_id, width, height;
{
    INSTANCE_HEADER *new;

    if ((new = (INSTANCE_HEADER *) malloc (sizeof (INSTANCE_HEADER))) == NULL)
        AllocError ("CreateInstanceHeader");
    new->instance_id = instance_id;
    new->width = width;
    new->height = height;
    new->entree = entree;
    /* add to instance list */
    new->next = instances;
    instances = new;
    /* fill in entry instance pointer */
    entree->instance = new;

    return (new);
}

FreeInstances()
{
    INSTANCE_HEADER *inst, *next;

    inst = instances;
    while (inst != NULL) {
        InstanceOfEntry(InstanceEntry(inst)) = NULL;
        next = NextInstance(inst);
        free(inst);
        inst = next;
    }
    instances = NULL;
}

```