

## Project 2 - C

For this project we will be writing a program that keeps track of objects on a map.

1. Create a header file called `mapobject.h` which defines the following:
  - a. A user defined enumeration type called `object_type_t` that has as its range of values `CAR`, `TREE`, `POLICE`, `OBSTACLE`, `EMPTY`, and any other features you want to define.
  - b. A user defined type called `object_t` that contains the following fields:
    - A string field called `label`
    - Two integer fields for coordinates called `xloc` & `yloc`
    - A float field for the speed
    - An int field for direction
    - The type of object (`object_type_t`)For example: `{"Car1", 1, 1, 55.0, 0, CAR}`.
  - c. A symbolic constant `LABEL_LEN` that indicates the size of your `object_t` label field.
  - d. Any forward declarations required for your methods
2. Create a C file called `mapobject.c` that defines the following functions:
  - a. `void get_object(object_t *)` prompts the user for information to fill a map object record.
  - b. `void print_object(const object_t *)` displays the record.
  - c. `char get_map_representation(object_type_t type)` that takes an `object_type_t` as a parameter and returns a character representation for that type of object
  - d. `object_type_t string_to_type(const char * type_string)` that takes a C string as a parameter and returns the corresponding `object_type_t`.
3. Define the following functions in the appropriate module:
  - a. `void print_objects(const node_t *)` prints a list of all objects
  - b. `void find_all_of_type(const node_t *, object_type_t)` prints the label, location, direction & speed of (all) the object(s) with the given type
  - c. `void find_object(const node_t *, const char *)` prints the location & speed of the object(s) with the given label
4. Complete the provided C program called `project2.c` which reads objects from `objects.txt` & stores the objects in the `basiclist` data structure. Add at least 2 additional objects to this database before the map is displayed in the main method. **Print your name at the start and end of your programs output.**

You should use good design principles, including using header/source files and deciding on which data types to use for the data members of object type. Use whitespace, comments, and good variable names to improve readability. Your output should be neat and concise. Feel free to add additional c/h files as you find appropriate, however you should not modify the `basiclist` or `map` files. Include a comment with your name at the top of each source file. See sample output on the next page. Your code should compile with gcc using the flags used in class.

Submit the **zipped source code** (the directory structure is up to you) including **all files required to compile and run** your program. Your submission should include at least the following files:

- `project2.c`
- `mapobject.c` / `mapobject.h` (your map object module)
- `basiclist.c` / `basiclist.h` (you should **not** modify these)
- `map.c` / `map.h` (you should **not** modify these)
- Any other C files you created (optional)
- A makefile (optional – I will provide a sample one)

In addition to your zipped source, **upload a separate Word, .txt or PDF file** with a brief description of your implementation (including any issues you ran into and how you resolved them) along with the contents of your `mapobject.h` header file, your `get_object()` method, and your `find_all_of_type` method().

Be prepared to demo your programs in class on the due date. **Start early!**

Sample Output (user input is bold):

```
$ ./project2.exe
ENTRY : Label: RedCar Location: (10,12) Speed: 65.00 Direction: 0 Type: CAR
ENTRY : Label: BlueCar Location: (14,5) Speed: 45.00 Direction: 270 Type: CAR
ENTRY : Label: Bus Location: (8,10) Speed: 55.00 Direction: 90 Type: CAR
ENTRY : Label: Police1 Location: (18,10) Speed: 55.00 Direction: 180 Type: POLICE
ENTRY : Label: Tree1 Location: (22,2) Speed: 0.00 Direction: 180 Type: TREE
ENTRY : Label: Tree2 Location: (4,8) Speed: 0.00 Direction: 180 Type: TREE
ENTRY : Label: Tree3 Location: (12,12) Speed: 0.00 Direction: 180 Type: TREE
ENTRY : Label: Tree4 Location: (14,8) Speed: 0.00 Direction: 180 Type: TREE
ENTRY : Label: Mattress1 Location: (20,12) Speed: 0.00 Direction: 180 Type: OBSTACLE
```

Printing map:

```
+---+---+---+---+---+---+
|   |   |   |   |   |   |
|   |   |   |   | *  |   |
|   |   |   |   |   |   |
+---+---+---+---+---+---+
|   |   |   | 0  |   |   |
|   |   |   |   |   |   |
| *  |   | *  |   |   |   |
|   |   |   |   |   |   |
+---+---+---+---+---+---+
|   | 0  |   | #  |   |   |
|   |   | 0  | *  |   | X  |
|   |   |   |   |   |   |
+---+---+---+---+---+---+
```

print\_objects():

```
Label: Mattress1 Location: (20,12) Speed: 0.00 Direction: 180 Type: OBSTACLE
Label: Tree4 Location: (14,8) Speed: 0.00 Direction: 180 Type: TREE
Label: Tree3 Location: (12,12) Speed: 0.00 Direction: 180 Type: TREE
Label: Tree2 Location: (4,8) Speed: 0.00 Direction: 180 Type: TREE
Label: Tree1 Location: (22,2) Speed: 0.00 Direction: 180 Type: TREE
Label: Police1 Location: (18,10) Speed: 55.00 Direction: 180 Type: POLICE
Label: Bus Location: (8,10) Speed: 55.00 Direction: 90 Type: CAR
Label: BlueCar Location: (14,5) Speed: 45.00 Direction: 270 Type: CAR
Label: RedCar Location: (10,12) Speed: 65.00 Direction: 0 Type: CAR
```

find\_all\_of\_type(CAR):

```
Bus : (8, 10) -> 90 @55.00
BlueCar : (14, 5) -> 270 @45.00
RedCar : (10, 12) -> 0 @65.00
```

find\_object(RedCar):

```
RedCar : (10, 12) -> 0 @ 65.00
```