

## 06 PREPARE : CHECKPOINT A - MAKEFILES

After completing (or *while* completing) the [preparation material](#) for this week, complete the following exercise.

### Overview

This checkpoint is intended to help you practice the syntax of separate compilation and makefiles. You will be provided a simple class and asked to prepare a makefile to perform the separate compilation and the linking required to produce a final executable.

### Instructions

Begin with a working implementation of a robot and point class from last weeks team activity. You can copy this code to your directory via the following:

```
mkdir check06a
cd check06a

cp /home/cs165new/check06a/* .
```

This code contains an empty makefile (with the header information you need to fill out). Your task is to open this makefile (e.g., "emacs makefile") and add the necessary rules to compile the executable.

Please make note of the following instructions / requirements:

1. The implementation of the robot class is in robot.cpp
2. The implementation of the point class is in point.cpp
3. The main function is in check06a.cpp
4. You must set up your makefile to compile these separately. In other words, your makefile must create robot.o, point.o, and check06a.o first.
5. Then, your make file should link all of the .o files together to produce a final executable (a.out).
6. Each rule in the makefile should list all of its dependencies. Test this by trying the following cases (plus more):
  - Change check06a.cpp, this should update check06a.o and a.out

- Change robot.cpp, this should update robot.o and a.out
  - Change robot.h, this should update robot.o, check06a.o, and a.out
7. Please note that the makefiles we have been using to this point (for team activities and prior assignments) DO NOT use separate compilation. They simply list all of the .cpp files in the same "g++" command. This is NOT sufficient for this assignment.

## Sample Output

The following is an example of output for this program:

```
Robot details: (5, 5) - Energy: 100
```

## Testbed

An auto-grading testbed script is provided for you to help evaluate your program. This same testbed script will be used to grade your program. It is pass/fail, so your program must pass the testbed completely for you to receive credit for this assignment. You may run the testbed as many times as you like.

Please note that it is possible to do this assignment without a makefile that adheres to all of the requirements above, but this will not help you understand the concept, and you will not receive credit if you do not follow the instructions.

## Helpful commands

```
tar -cf check06a.tar makefile *.h *.cpp
testBed cs165new/check06a check06a.tar
submit check06a.tar
```