

```

Jan 12, 16 11:18                make_area                Page 1/3
SHELL=/bin/sh

# Note: Comments start with #. $(FOOBAR) means: evaluate the variable
#       defined by FOOBAR= (something).

# This file contains a set of rules used by the "make" command.
# This makefile $(MAKEFILE) tells "make" how the executable $(COMMAND)
# should be create from the source files $(SRCS) and the header files
# $(HDRS) via the object files $(OBS); type the command:
#     "make -f make_program"
# where make_program should be replaced by the name of the makefile.
#
# Programmer: Dick Furnstahl (furnstahl.1@osu.edu)
# Latest revision: 12-Jan-2016
#
# Notes:
# * If you are ok with the default options for compiling and linking, you
#   only need to change the entries in section 1.
#
# * Defining BASE determines the name for the makefile (prepend "make_"),
#   executable (append ".x"), zip archive (append ".zip") and gzipped
#   tar file (append ".tar.gz").
#
# * To remove the executable and object files, type the command:
#   "make -f $(MAKEFILE) clean"
#
# * To create a zip archive with name $(BASE).zip containing this
#   makefile and the SRCS and HDRS files, type the command:
#   "make -f $(MAKEFILE) zip"
#
# * To create a gzipped tar file with name $(BASE).tar.gz containing this
#   makefile and the source and header files, type the command:
#   "make -f $(MAKEFILE) tarz"
#
# * Continuation lines are indicated by \ with no space after it.
#   If you get a "missing separator" error, it is probably because there
#   is a space after a \ somewhere.
#
#####
# 1. Specify base name, source files, header files, input files
#####

# The base for the names of the makefile, executable command, etc.
BASE= area

# Put all C++ (or other) source files here. NO SPACES after continuation \'.
SRCS= \
area.cpp

# Put all header files here. NO SPACES after continuation \'.
HDRS= \

# Put any input files you want to be saved in tarballs (e.g., sample files).
INPFILE= \

#####
# 2. Generate names for object files, makefile, command to execute, tar file
#####

# *** YOU should not edit these lines unless to change naming conventions ***

OBS= $(addsuffix .o, $(basename $(SRCS)))
MAKEFILE= make_$(BASE)
COMMAND= $(BASE).x
TARFILE= $(BASE).tar.gz
ZIPFILE= $(BASE).zip

#####

```

Tuesday January 12, 2016

make_area

```

Jan 12, 16 11:18                make_area                Page 2/3
# 3. Commands and options for different compilers
#####
#
# Compiler parameters
#
# CXX          Name of the C++ compiler to use
# CFLAGS      Flags to the C++ compiler
# CWARNS      Warning options for C++ compiler
# F90         Name of the fortran compiler to use (if relevant)
# FFLAGS      Flags to the fortran compiler
# LDFLAGS     Flags to the loader
# LIBS        A list of libraries
#
CXX= g++
CFLAGS= -g -O2
CWARN= -Werror -Wall -W -Wshadow -fno-common
MOREFLAGS= -Wpedantic -Wpointer-arith -Wcast-qual -Wcast-align \
           -Wwrite-strings -fshort-enums

# add relevant libraries and link options
LIBS=
LDFLAGS= -lgs1 -lgs1cblas

#####
# 4. Instructions to compile and link, with dependencies
#####
all: $(COMMAND)

.SUFFIXES:
.SUFFIXES: .o .mod .f90 .f .cpp

%.o: %.mod

# This is the command to link all of the object files together.
# For fortran, replace CXX by F90.
$(COMMAND): $(OBS) $(MAKEFILE)
$(CXX) -o $(COMMAND) $(OBS) $(LDFLAGS) $(LIBS)

# Command to make object (.o) files from C++ source files (assumed to be .cpp).
# Add $(MOREFLAGS) if you want additional warning options.
.cpp.o: $(HDRS) $(MAKEFILE)
$(CXX) -c $(CFLAGS) $(CWARN) -o $@ $<

# Commands to make object (.o) files from Fortran-90 (or beyond) and
# Fortran-77 source files (.f90 and .f, respectively).
.f90.mod:
$(F90) -c $(F90FLAGS) -o $@ $<

.f90.o:
$(F90) -c $(F90FLAGS) -o $@ $<

.f.o:
$(F90) -c $(FFLAGS) -o $@ $<

#####
# 5. Additional tasks
#####

# Delete the program and the object files (and any module files)
clean:
/bin/rm -f $(COMMAND) $(OBS)
/bin/rm -f $(MODIR)/*.mod

# Pack up the code in a compressed gnu tar file
tarz:
tar cfvz $(TARFILE) $(MAKEFILE) $(SRCS) $(HDRS) $(MODIR) $(INPFILE)

```

1/2

```
# Pack up the code in a zip archive
zip:
    zip -r $(ZIPFILE) $(MAKEFILE) $(SRCS) $(HDRS) $(MODIR) $(INPFILE)
#####
# That's all, folks!
#####
```