```
Name:     Robert S. Bauer
UANET id: rsbauer

finddups synax check

finddups
./HistoryPaper2.txt
./old/Paper2.txt
./HistoryPaper1.txt

finddups --long
./old/Paper1.txt
./HistoryPaper2.txt

./old/save/Notes.txt
./old/Paper2.txt
./HistoryPaper1.txt
```

```bash
#!/bin/bash

##
#
# Rob Bauer       01-OCT-2011
# Script to find duplicates, preserving the oldest dated file and handling
# multiple duplicate files.
#
# Example usage:
#
#       - find duplicates in current and subdirectories
#       finddups
#
#       - find duplicates in different directory and its subdirectories
#       finddups [path]
#
#       - list all files that where duplicated
#       finddups --list
#
#       - remove files (script will *NOT* prompt user to delete each one)
#   - use at your own risk
#       finddups --rm
#
# Known issues:
#       Script does not use cmp to do a direct file comparison and is strictly
#       based on md5sum which may produce hash collisions and false duplicate
#       matches.
#
# This script is a for/while loop free.
#
##

# set the default dir to search from, if not specified on the command line
defaultdir=.
isRemove=0
isList=0
SCRIPTDIR="$( cd -P "$( dirname "$0" )" && pwd )"

# process command line args
while [ "$1" != "" ]; do
        case $1 in
                --long )
                                        isList=1
                                        ;;
                --rm )
                                        isRemove=1
                                        ;;
                * )
                                        defaultdir=$1
                                        ;;
        esac
        shift
done

# check if the defaultdir is actually a file - if its a file, calc the hash
# and date
if [ -f "$defaultdir" ] ; then
        # calc the md5sum
        md5hash=‘md5sum "$defaultdir" | cut -d ’ ’ -f1‘

        # get the file date and time
        filedate=‘find $defaultdir -not -empty -type f -printf "%TY%Tm%Td%TH%TM%TS"
‘

        echo "$md5hash $filedate $defaultdir"

        exit 0
fi

# get a list of files once to use throughout the script
# pipes back to itself to get the md5sum hash, file date, and file name on one
# line so the sort can find the matching hash AND oldest file (so it can be
# kept)
filelist=‘find $defaultdir -not -empty -type f -print0 | xargs -0 -I% "$SCRIPTDIR"/
finddups % | sort‘

# check if user wanted full output
if [ "$isList" = "1" ]; then
        # show a list of all files, separating the different sets of duplicates
        echo "$filelist" | uniq -w32 -d --all-repeated=separate | cut -d ’ ’ -f3
else
        # show a list of duplicate files
        keeplist=‘echo "$filelist" | uniq -w32 -d‘
        duplist=‘echo "$filelist" | uniq -w32 -D | grep -v "$keeplist" | cut -d ’ ’
 -f3‘
        echo "$duplist"

        # remove files
        if [ "$isRemove" = "1" ]; then
                # add "-pr" to have rm delete with prompting
                echo "$duplist" | xargs -I% rm %
        fi
fi
```

```
class synax check
```

```bash
#!/bin/bash

##
#
# Rob Bauer             01-OCT-2011
# Utility functions for managing svn repositories used in a class setting
#
##

## ## ##   VARIABLES   ## ## ##

accountid=
labnumber=
syncdir=

## ## ##   DIR/FILE VARIABLES   ## ## ##

SCRIPTDIR="$( cd -P "$( dirname "$0" )" && pwd )"
BASEDIR=${SCRIPTDIR%/*/*}
CURRENTDIR=‘pwd‘
CURRENTSUBDIR=${CURRENTDIR#"$BASEDIR"}
COURSE=‘echo "$CURRENTSUBDIR" | cut -d ’/’ -f2‘
SECTION=‘echo "$CURRENTSUBDIR" | cut -d ’/’ -f3‘
accountfilename=admin/accounts.txt
adminfilename=admin/admin.txt
```

```
templatedir="$BASEDIR/$COURSE/template/"


## ## ##   FUNCTIONS   ## ## ##

showHelp()
{
        echo "
Utility functions for managing svn repositories used in a class setting
Rob Bauer
01-OCT-2011

-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

Display help information:
class help


Output the current course:
class course



Output the current section:
class section


Output the account ids of all students in this section:
class accounts


Output the account ids of the admin accounts:
class --admin accounts


Create the folder for the account <accountid> using the template folder as a guide.
 If the folder already exists, the files will not be overwriten.
class create <accountid>


Create the directory for all the accounts in this section using the template folder
 as a guide. If the folder already exists, the files will not be overwritten.
class create


Synchronize account <accountid> with template folder. Note that if a file/directory
 already exists, the sync will not overwrite it.
class sync <accountid>


Synchronize account <accountid> with template path <dir>:
class sync <accountid> <dir>


Create the lab report LABNUMBER for accountid:
class labreport <LABNUMBER> <accountid>


Create the lab report LABNUMBER for all accounts in this section:
class labreport <LABNUMBER>


Print to a PDF file the lab report LABNUMBER for accountid:
class print <LABNUMBER> <accountid>


Print to a single PDF file the lab reports LABNUMBER for all accounts in this secti
on:
class print <LABNUMBER>
"
}

showCourse()
{
        if [ -z "$COURSE" ]; then
                echo "Currently not located within a course. Please change director
y to a course."
        else
                echo "$COURSE"
        fi
}

showSection()
{
        if [ -z "$SECTION" ]; then
                echo "Currently not located within a section. Please change directo
ry to a section."
        else
                echo "$SECTION"
        fi
}

showAccounts()
{
        accountfile="$BASEDIR/$COURSE/$SECTION/$accountfilename"
        if [ -e "$accountfile" ]; then
                cat "$accountfile"
        fi
}

showAdminAccounts()
{
        accountfile="$BASEDIR/$COURSE/$SECTION/$adminfilename"
        if [ -e "$accountfile" ]; then
                cat "$accountfile"
        fi
}

createAccount()
{
        # check if account dir exists
        accountdir="$BASEDIR/$COURSE/$SECTION/students/$accountid"

        if [ ! -e "$accountdir" ]; then
                mkdir "$accountdir"
        fi

        # a=archive (preserve permissions, etc),v=verbose,z=compress
        rsync -avz "$templatedir" "$accountdir"
}

createAllAccounts()
{
        class accounts | xargs -I% class create %
}

checkCourseSection()
{
                # check that there is a course
        if [ "$COURSE" = "" ]; then
                echo "Course is required"
                exit 1
        fi

        # check that there is a section
        if [ "$SECTION" = "" ]; then
                echo "Section is required"
                exit 1
        fi
}

syncAccount()
{
        accountdir="$BASEDIR/$COURSE/$SECTION/students/$accountid"

        # a=archive (preserve permissions, etc),v=verbose,z=compress
        rsync -avz --ignore-existing "$syncdir" "$accountdir"
}

createLabreport()
{
        accountdir="$BASEDIR/$COURSE/$SECTION/students/$accountid"
        labdir="$accountdir/lab$labnumber"
        plugindir="$BASEDIR/$COURSE/labplugin"

        if [ -e "$labdir" ]; then
                echo "$accountid:"
                `$plugindir/lab$labnumber $accountdir $labnumber`
        fi
}

createAllLabReports()
{
        class accounts | xargs -I% class labreport "$labnumber" %
}

printLabreport()
{
        accountdir="$BASEDIR/$COURSE/$SECTION/students/$accountid"
        labdir="$accountdir/lab$labnumber"
        plugindir="$BASEDIR/$COURSE/labplugin"

        if [ -e "$labdir" ]; then
                echo "$accountid:"
                $plugindir/printlab $accountdir $labnumber
        fi
}

printAllLabReports()
{
        class accounts | xargs -I% class print "$labnumber" %
}

## ## ##   PARSE ARGS   ## ## ##

# process command line args
while [ "$1" != "" ]; do
        case $1 in
                help )
                                        showHelp
                                        exit 0
                                        ;;
                course )
                                        showCourse
                                        exit 0
                                        ;;
                section )
                                        showSection
                                        exit 0
                                        ;;
                accounts )
                                        showAccounts
                                        exit 0
                                        ;;
                --admin )
                                        showAdminAccounts
                                        exit 0
                                        ;;
                create )
                                        checkCourseSection

                                        # user may be trying to create all accounts
or just one

                                        shift
                                        if [ "$1" != "" ]; then
                                                accountid="$1"
                                                createAccount
                                        else
                                                createAllAccounts
                                        fi

                                        exit 0
                                        ;;

                sync )
                                        checkCourseSection

                                        # grab accountid
                                        shift
                                        if [ "$1" != "" ]; then
                                                accountid="$1"
                                                syncdir="$templatedir"
                                                shift

                                                # check if dir was specified
                                                if [ "$1" != "" ]; then
                                                        syncdir="$1"
                                                fi

                                                syncAccount
                                        fi

                                        exit 0
                                        ;;
                labreport )
                                        checkCourseSection

                                        # grab labnum
                                        shift
                                        if [ "$1" != "" ]; then
                                                labnumber="$1"
                                                syncdir="$templatedir"
                                                shift

                                                # check if account was specified
                                                if [ "$1" != "" ]; then
                                                        accountid="$1"
                                                        createLabreport
                                                else
                                                        createAllLabReports
                                                fi

                                        fi

                                        exit 0
                                        ;;
                print )
                                        checkCourseSection

                                        # grab labnum
                                        shift
                                        if [ "$1" != "" ]; then
                                                labnumber="$1"
                                                syncdir="$templatedir"
                                                shift
                                                # check if account was specified
                                                if [ "$1" != "" ]; then
                                                        accountid="$1"
                                                        printLabreport
                                                else
                                                        printAllLabReports
                                                fi

                                        fi

                                        exit 0
                                        ;;

                * )
                                        echo "Command not found. Enter class help f
or a list of commands."

                                        ;;
        esac
        shift
done
```