# This script first prints a line asking the user to input their name, then reads the users input from the command line and prints out a string containing the name

```
#!/bin/bash
echo “Please enter your name, followed by [ENTER]:”
read name
echo “Your name is $name”
```

# This script defines a variable called “file” and assigns it the path to a file. It then reads the file line by line and prints out each line.

```
#!/bin/bash
file=”./name-script.sh”
while read line                 # The while loop uses the “read” command which
do                              ,by default, uses the newline character “\n” a
     echo “$line”               delimiter
done < “$file”                  # The “< $file” defines what the while loop
                                should loop over
```

# This script accepts a file as a command-line argument, read the file line by line, and prints each line.

```
#!/bin/bash
while read line
do
     echo “$line”
done < $1                       # Remember, we can address fields with
                                $<number>
```

# This script accepts a file as a command-line argument, counts the number of lines if they are not blank, and prints the total number of lines.

```
#!/bin/bash
while read line; do                 # Semicolon is only needed when the end of
     if [ “$line” != “” ]; then     a line is missing (i.e., when two statements
      ((count++))                   are on the same line)
          echo “$line”
     fi
done < $1
echo $count
```

This script accepts a file as a command-line argument, but instead of assigning the whole line to a variable, it assigns different portions of the line to different variables. By default, read use newline and whitespace as a delimiter.

```
#!/bin/bash
while read first second third rest; do
     echo “First word is $first, second word is $second, third
          word is $third, and the rest of line is $rest”
done < $1
```