

## Lab: Operations

When we create a new person, we need to get an unused ID to assign. Let's create a couple of functions to generate new IDs for us.

1. Open your library of customer functions.
2. Create a new function called getRandomID(). Have it return some number – any number for now.
3. Create a test script that includes your functions and calls getRandomID(). Have it print out the number that is being returned.
4. Run and test. Do you see the number? Cool. Let's move on.

### Making it return a useful number

We don't want just any number, we want a number that isn't already being used. Let's make that happen.

5. Create a variable called ISUSED. Set it equal to "true".
6. Write a loop like so:

```
while true ; do
  # Generate a RANDOM Id
  # See if the number is already used
  if [[ $ISUSED = "true" ]] ; then
    echo "Rats, $NEWID is used."
  else
    echo "Yay! $NEWid isn't used."
    break
  fi
  sleep 1
done
```

7. Run and test. It should loop forever printing that the number is used. When it does, you can move on.
8. Where it says to generate a RANDOM id, make it do that. Give us a number between 1 and 10,000 and store it in \$NEWID.
9. Where it says to see if the number is already used, do that also. See if that number appears in the ID column of your customers file. If not, set ISUSED to false. (hint: use *grep*)
10. Have it return the unused random number.
11. After you get it working, remove all of your echoes and sleeps.

### Getting the next id

Getting a random number will be great, but it can be very, very slow when we get a lot of customers. We should only call it as a last resort; if we run out of numbers. The quicker way would be to get the next biggest number.

12. Create a new function called getNextID().

13. In it, create a variable called BIGGESTID and set it equal to zero (0).
14. In it, loop through your customer file sort of like you did in the script where you read all the customers:

```
for LINE in $(cat $CUSTOMERFILE) ; do
    print $LINE
    # Get the ID
    # Reassign BIGGESTID?
done
```

15. Run and test, making sure you can see each line.
16. Pull out the ID from the line. (Hint: use *cut*)
17. Compare the current ID to the prior biggest. If this ID is bigger, set BIGGESTID to this one.
18. After the loop, add 1 to the biggest ID and save it as NEWID.
19. Return the new ID.
20. Remove any unneeded debugging statements like the printing of \$LINE.