

Points: 100

Due Date: Friday, November 9 @ 11:59pm

This is an individual assignment.

## Task:

Write a Java project that:

- Implements the cyclic-shift hash code computation described in the text.
- Calculates and displays the cyclic-shift hash codes for the strings:
  - o POTS
  - o STOP
  - o TOPS
  - o This set of strings should be hard-coded into your program.
- Implements a verbose version of the cyclic-shift hash code computation that shows the 32-bit integer bit pattern at each stage of the cyclic-shift hash code computation ( see example below ).
- Displays the verbose cyclic-shift hash code computation for the above strings.

## Turning in your assignment:

- **Make sure that all of your code is properly documented.**
- Turn in your assignment using the standard method.
- Create a Word document:
  - o Copy and paste each of your Java files into the document.
  - o Paste the contents of the terminal showing the complete output of a complete run of your program after the Java code in your document.
- Export your NetBeans project to a zip archive.
- Turn in the Word document and zipped project as to separate files in a single Blackboard submission.

## Example Output ( only shows verbose for one string ):

Summary hash code information:

```
POTS : 00000000001010010100011011010011
STOP : 00000000001010101101101000110000
TOPS : 00000000001010110100011001010011
```

Detailed hash code information:

Creating hash code for POTS:

```
Entering hashCode, pass 0      00000000000000000000000000000000
```

```
hashCode <<5                  00000000000000000000000000000000
```

```
hashCode >>> 27               00000000000000000000000000000000
```

```
hashCode <<5 | hashCode>>>27 00000000000000000000000000000000
```

```
Adding Character P            0000000000000000000000000001010000
```

```
Exiting hashCode              0000000000000000000000000001010000
```

```
Entering hashCode, pass 1     0000000000000000000000000001010000
```

```
hashCode <<5                  00000000000000000000000000010100000000
```

```
hashCode >>> 27               000000000000000000000000000000000000
```

```
hashCode <<5 | hashCode>>>27 00000000000000000000000000010100000000
```

[illegible]