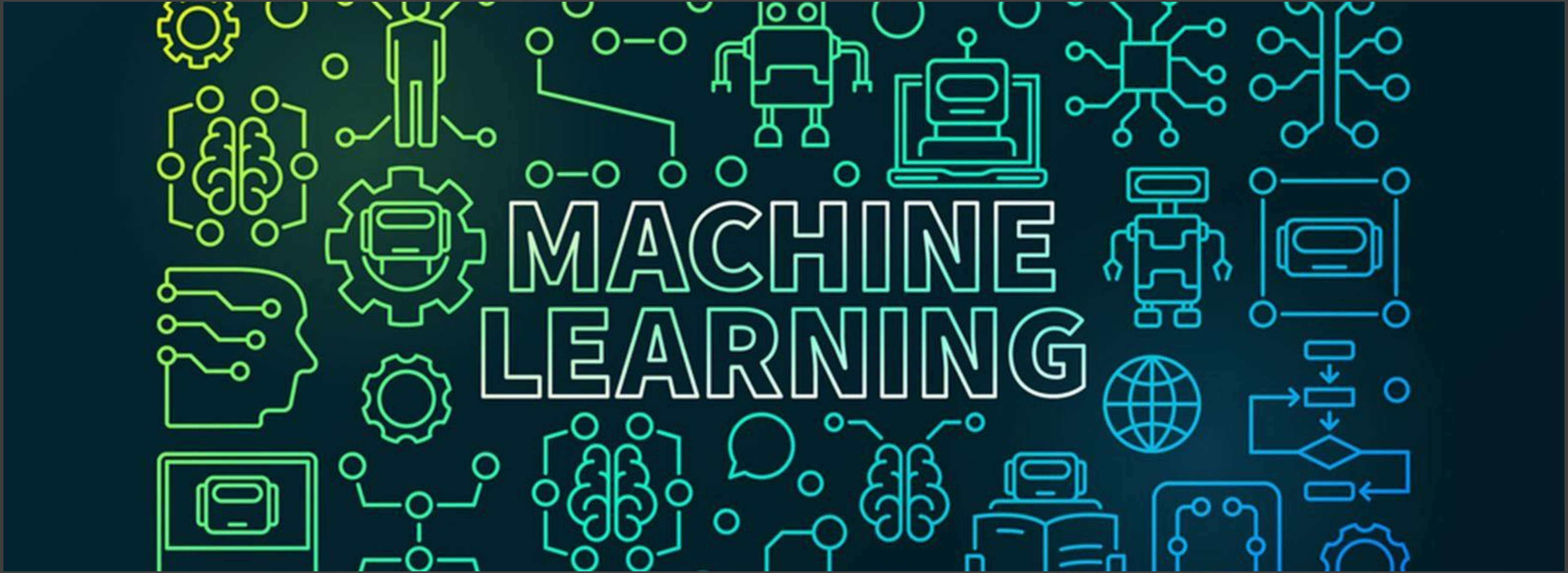


Data Science
Programming: ISM6251

Malware Detection

By: Suryateja Ch.

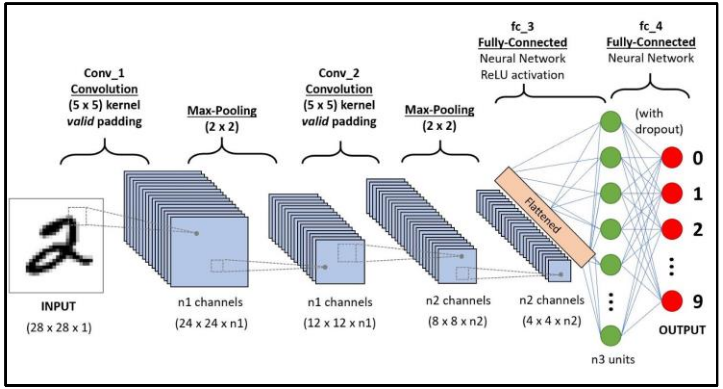
Approach

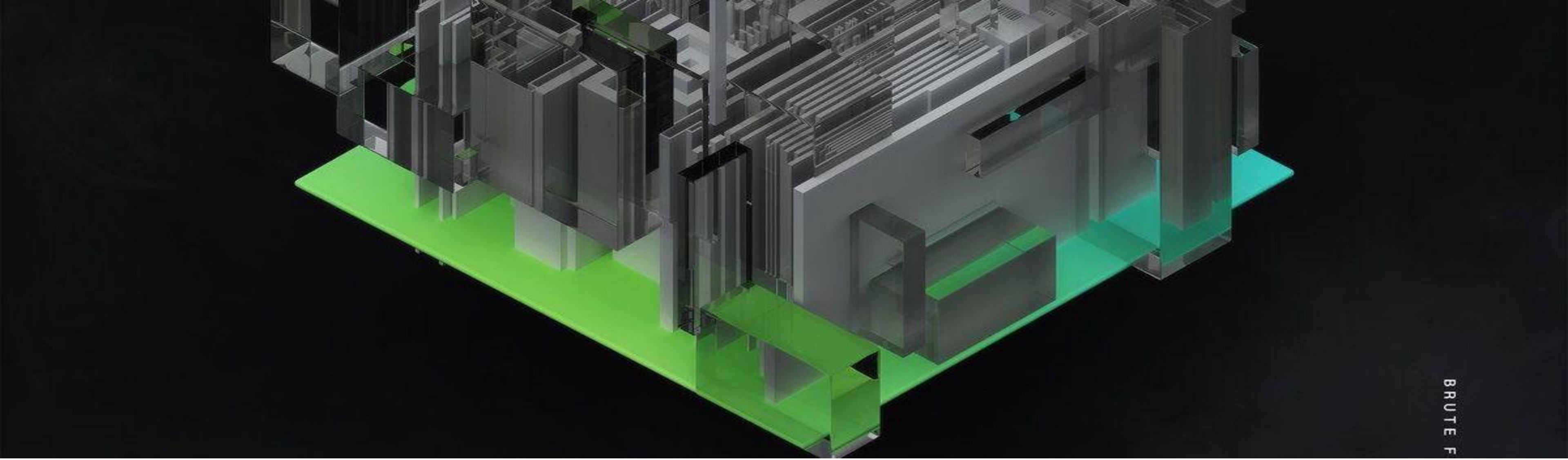
- In The Past Few Years, Malware Attacks Has Grown Very Rapidly. The Syndicates That Invest Heavily In Technologies To Evade Traditional Protection.
- The Major Part Of Protection For A Computer System Is From A Malware Attack Where We Need To Identify Whether A Given File/Software Is Safe.
- Given An Input Of Common Executables, Classify If The File Belongs To The Malware Class.

```
Student me = Student();

while (me.alive()) {
    me.sleep();
    continue;
    me.eat();
    me.practice();
    me.work();
    me.makeacontributiontosociety();
    me.beproductive();
    me.doliterallyanything();
}
```

Data - Source

- Data Sourced From: https://www.malwaredatascience.com/code-and-data
- Dataset Consists Of About 1500 Objects.
- There Are 9 Types Of Malwares In Our Training Data.
- We Found That Our Data Is balanced Where Class Occurrence Is almost Some.





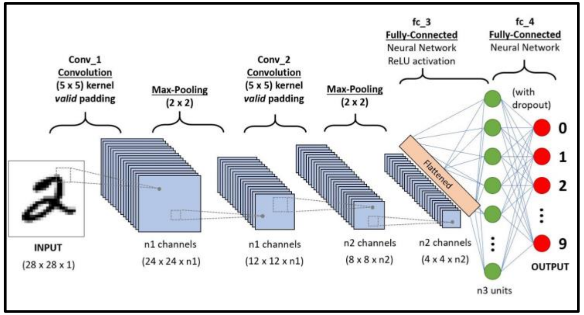
Objectives

Predict The Probability Of Each Data Point Belonging To Each Of The Nine Classes

Constraints

- Use Multiclass Probability Estimate
- Malware Detection Should Be Quick And Resource Optimized

Performance Metric



- Confusion Matrix:
- Multi Class Log Loss:

D ={x_i , y_i } I =I to N

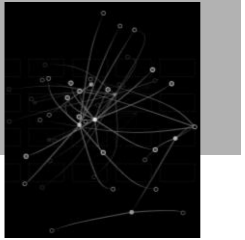
MCLL= $$-\frac{1}{N}\sum_{i}^{N}\sum_{j}^{M} y_{ij} \cdot Ln(p_{ij}))$$

yij=I   if x_i € class j   yij is our ground truth
else 0

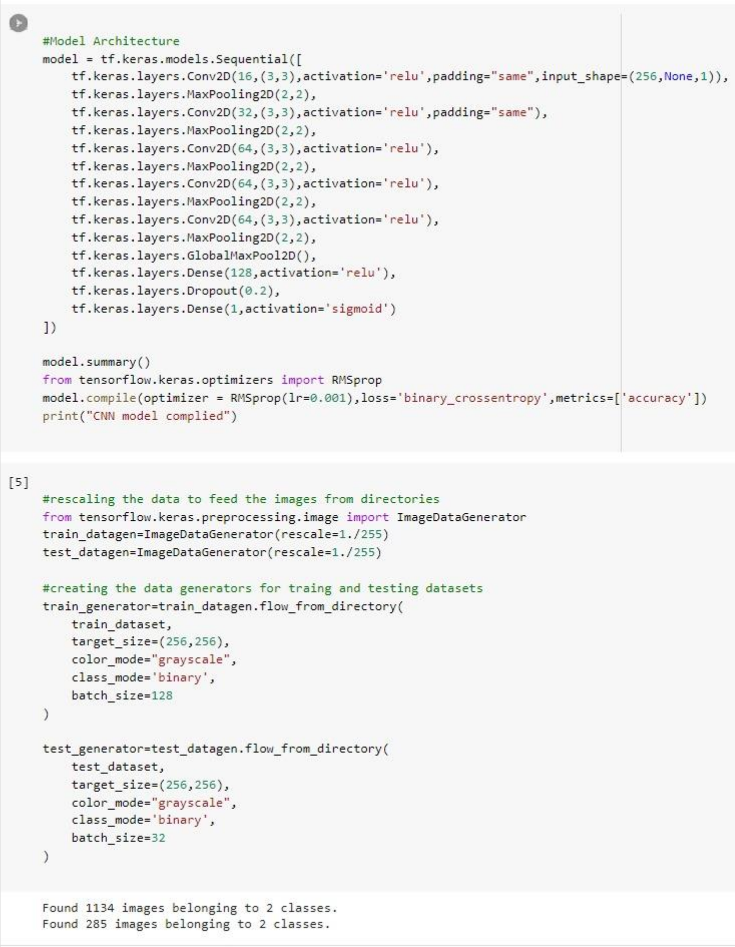
Pij= probability(xi € class j)   Pij is model predicted value

Progress & Future Scope

- To Solve This Problem, We Are Using An Index Minimizing Framework.
- So Far, We Have Completed Exploratory Data Analysis And Class Distribution.
- With CNN The Accuracy Is 94% For 2-Class. Experimenting With Other Keras Models Like VGG16, ResNet etc. and TensorFlow.
- Expand The Scope To Other Types Of Files Type Like .Txt, .Xls, .Doc, .Pdf Etc.
- Convert Input To Video Instead Of Images.



Code Snippet & Libraries Used

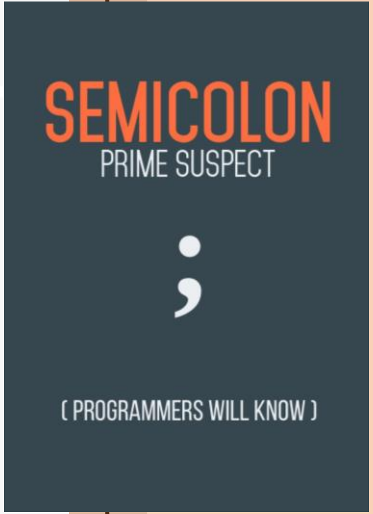


```
#Model Architecture
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16,(3,3),activation='relu',padding="same",input_shape=(256,None,1)),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(32,(3,3),activation='relu',padding="same"),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64,(3,3),activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64,(3,3),activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(64,(3,3),activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.GlobalMaxPool2D(),
    tf.keras.layers.Dense(128,activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(1,activation='sigmoid')
])

model.summary()
from tensorflow.keras.optimizers import RMSprop
model.compile(optimizer = RMSprop(lr=0.001),loss='binary_crossentropy',metrics=['accuracy'])
print("CNN model complied")
```

```
[5]
#rescaling the data to feed the images from directories
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255)
test_datagen=ImageDataGenerator(rescale=1./255)

#creating the data generators for traing and testing datasets
train_generator=train_datagen.flow_from_directory(
    train_dataset,
    target_size=(256,256),
    color_mode="grayscale",
    class_mode='binary',
    batch_size=128
)

test_generator=test_datagen.flow_from_directory(
    test_dataset,
    target_size=(256,256),
    color_mode="grayscale",
    class_mode='binary',
    batch_size=32
)
```

```
Found 1134 images belonging to 2 classes.
Found 285 images belonging to 2 classes.
```

- Tensorflow
- Keras
- Pandas, Numpy
- Matplotlib



- Input: Images Files
- Output: Classification
- Evaluation Metrics: Confusion Matrix, ROC, Precsion And Recall

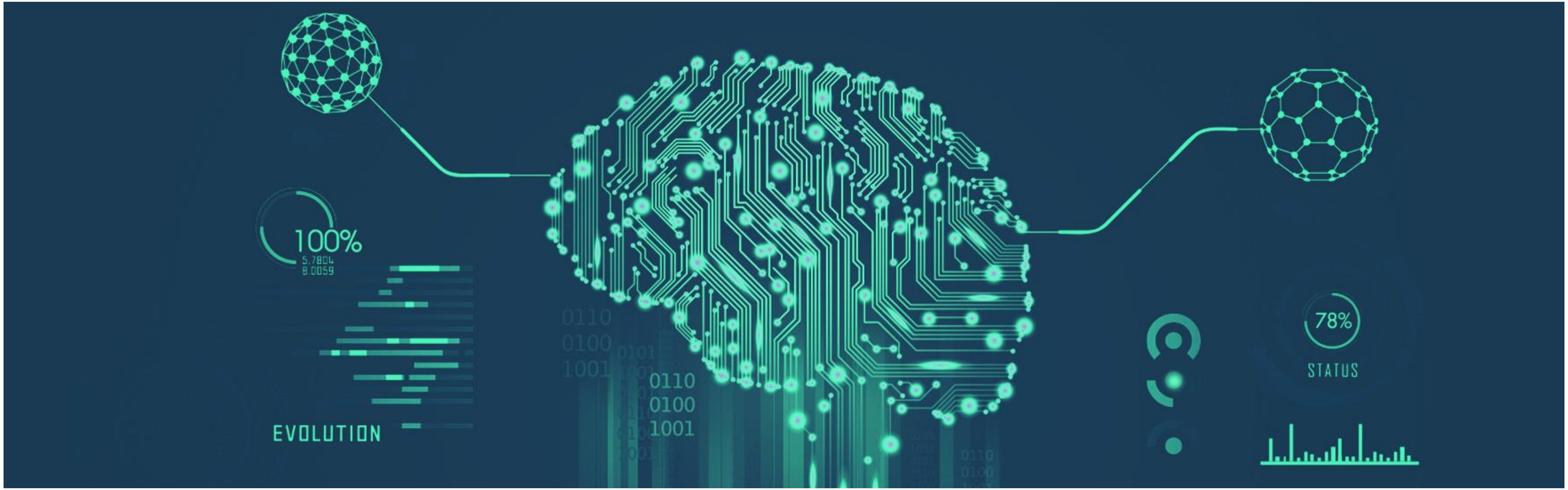
Applications: Broad & Niche

Broad Applications:

- Consumer Electronics
- Handhelds
- National Security
- Banking Applications

Niche Applications:

- IOT Devices
- Security Monitoring Systems
- Software Auditing



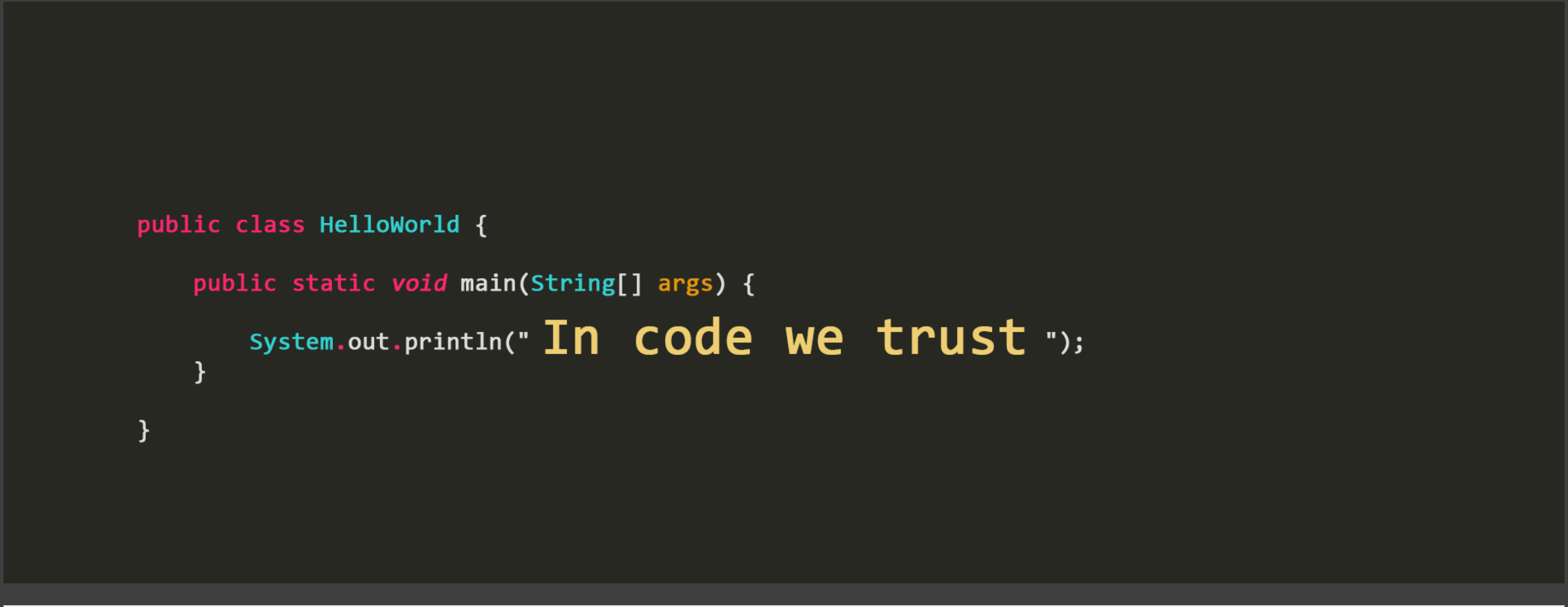
References

```
 *
 * @var boolean
 */
define('PSI_INTERNAL_XML', false);

if (version_compare("5.2", PHP_VERSION, ">")) {
    die("PHP 5.2 or greater is required!!!");
}
if (!extension_loaded("pcre")) {
    die("phpSysInfo requires the pcre extension to php in order to work
        properly.");
}

require_once APP_ROOT.'/includes/autoloader.inc.php';

// Load configuration
require_once APP_ROOT.'/config.php';

if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {
    $tpl = new Template("/templates/html/error_config.html");
    echo $tpl->fetch();
    die();
                                  javascript
                                        ; strtolower(
```

- Mallet, H. (2020, May 28). Malware Classification using Deep Learning - Tutorial | Towards Data Science. Medium. https://towardsdatascience.com/malware-classification-using-convolutional-neural-networks-step-by-step-tutorial-a3e8d97122f

- Rafique, M. F. (2019, October 24). Malware Classification using Deep Learning based Feature. . . ArXiv.Org. https://arxiv.org/abs/1910.10958

- Li, C. (2021, May 27). Journal of Cyber Security and Mobility. Riverpublishers. https://journals.riverpublishers.com/index.php/JCSANDM/article/view/6227



```
public class HelloWorld {

    public static void main(String[] args) {

        System.out.println(" In code we trust ");
    }

}
```

UNIVERSITY OF SOUTH FLORIDA

A PREEMINENT RESEARCH UNIVERSITY

Questions?

Thank You