SRI VASAVI ENGINEERING COLLEGE
PEDATADEPALLI, TADEPALLIGUDEM-534 101
Department of Computer Science & Engineering
(Accredited by NBA)

# SCUD ARSENAL

MAGAZINE

Volume-9
Issue-4


# Editors page

## Chief Editor:

Dr. D. Jaya Kumari ,
Professor and
Head of the Department.

## Contributing Editors:

1.Mrs.B.Sri Ramya
Asst.Professor , CSE.

2.Mr.N.V.M.K.Raja
Asst.Professor , CSE.

## Student Coordinators:

1.A.N.V.Vamsi Krishna
17A81A0501

2.G.B.S.Y.Vasudeva Rao
17A81A0516

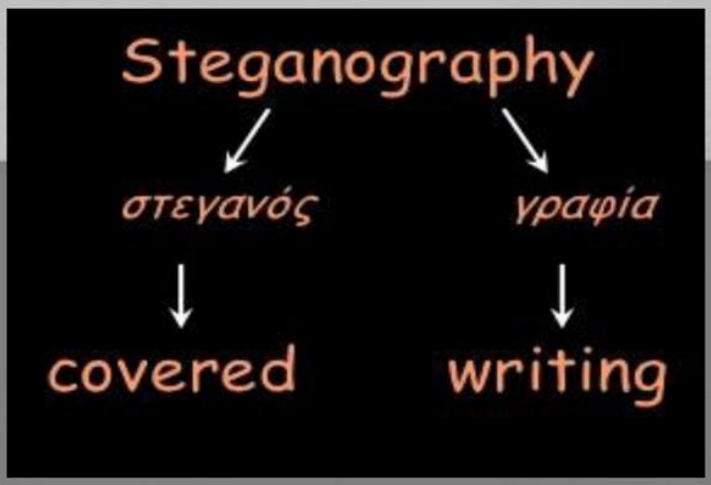
3. S. Mani Kumar
17A81A0539

4. P.Manasa
17A81A0506

# CONTENTS

# Scud Arsenal

## STEGANOGRAPHY




***Steganography is the art of hiding information in ways that prevent the detection of hidden messages.***

***It includes a vast array of secret communications methods that conceal the message's very existence.***

***In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol.***

***Media files are ideal for steganographic transmission because of their large size.***

***For example, a sender might start with an innocuous image file and adjust the color of every hundredth pixel to correspond to a letter in the alphabet. The change is so subtle that someone who is not specifically looking for it is unlikely to notice the change.***

**TECHNIQUES**

**1. PHYSICAL**

***. Invisible ink***

***.Tattoo message on head***

# Scud Arsenal

*. Overwrite select characters in printed type in pencil*

*. Pin punctures*

*. Microdots*

*. Newspaper clippings, Knitting instructions, XOXO signatures, report cards*

## 2. DIGITAL MESSAGES

*Modern steganography entered the world in 1985 with the advent of personal computers being applied to classical steganography problems*

*.Concealing messages within the lowest bits of noisy images or sound files.*

*. Concealing data within encrypted data or within random data.*

*.Changing the order of elements in a set*

*. Modifying the echo of a sound file.*

**Image of a tree with a steganographically hidden image. The hidden image is revealed by removing all but the two least significant bits of eachcolor component and a subsequent normalization. The hidden image is shown**

# Scud Arsenal

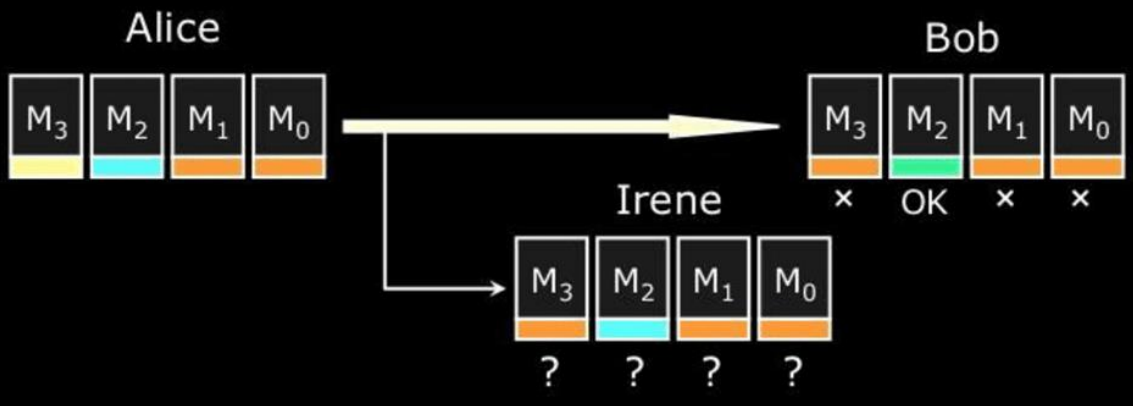
Image of a cat extracted

from the tree image above.

## 3. CHAFFING AND WINNOWING

### Chaffing & Winnowing

- Separate good messages from the bad ones
- Stream of unencoded messages with signatures
  - Some signatures are bogus
  - Need key to test

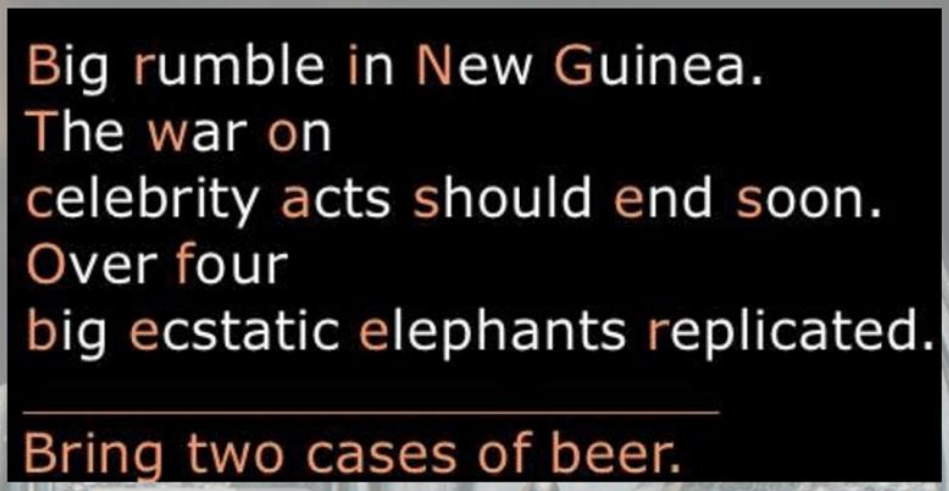


# Scud Arsenal

**4. NULL CIPHER**

**.Hide message among irrelevant data**

**. Confuse the crypto analyst**




**5. NETWORK STEGANOGRAPHY**

**Contrary to typical steganographic methods that use digital media (images, audio and video files) to hide data, network steganography uses communication protocols' control elements and their intrinsic functionality. As a result, such methods can be harder to detect and eliminate.**

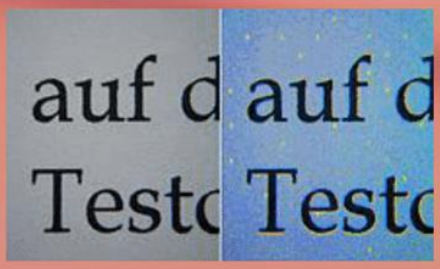
**6. CYBER PHYSICAL SYSTEMS**

**Specific techniques hide data in CPS components. For instance, data can be stored in unused registers of IoT/CPS components and in the states of IoT /CPS actuators**

**7. PRINTED**

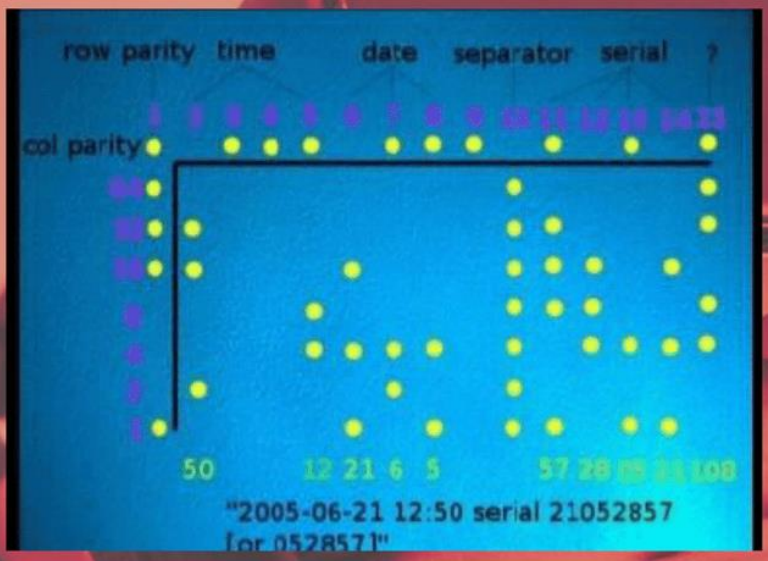
**Some types of modern color laser printers integrate the model, serial number and timestamps on each printout for traceability reasons using a dot-matrix code made of small, yellow dots not recognizable to the naked eye**

# Scud Arsenal




**Yellow dots from a laser printer**

**Machine ID codes in laser printers**




**The other techniques include video, audio, UV watermarking, text.**

**DETECTION**

**. Comparing them with previous files**

**. All files on suspect file system can be hashed using a hash function and then compared to hash table.**

**. A steganography tool can be used to camouflage the secret message in the least**

# Scud Arsenal

significant bits but it can introduce a random area that is too perfect. This area of perfect randomization stands out and can be detected by comparing the least significant bits to the next-to-least significant bits on image that hasn't been compressed.

**APPLICATIONS**

. Used in modern printers

. Allegedly used by terrorists

. Alleged use by intelligence services

. Digital watermarking

. Distributed steganography

. Online challenge

**BY**

**N.V.Sri Lalitha**

**17A81A0530**

# How to build a Neural Network with Keras

Keras is one of the most popular Deep Learning libraries out there at the moment and made a big contribution to the commoditization of artificial intelligence. It is simple to use and it enables you to build powerful Neural Networks in just a few lines of code. In this post, you will discover how you can build a Neural Network with Keras that predicts the sentiment of user reviews by categorizing them into two categories: positive or negative. This is called Sentiment Analysis and we will do it with the famous imdb review dataset. The model we will build can also be applied to other Machine Learning problems with just a few changes.

Note that we will not go into the details of Keras or Deep Learning. This post is intended to provide you with a blueprint of a Keras Neural Network and to make you familiar with its implementation.

**Table of Contents:**



## What is Keras?

Keras is an open source python library that enables you to easily build Neural Networks. The library is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, and MXNet. Tensorflow and Theano are the most used numerical platforms in Python to build Deep Learning algorithms but they can be quite complex and difficult to use. In comparison, Keras provides an easy and convenient way to build deep learning models. It's creator François Chollet developed it to enable people to build Neural Networks as fast and easy as possible. He laid his focus on extensibility, modularity, minimalism and the support of python. Keras can be used with GPUs and CPUs and it supports both Python 2 and 3. Google Keras made a big contribution to the commoditization of deep learning and artificial intelligence since it has commoditized powerful, modern Deep Learning algorithms that previously were not only inaccessible but also unusable as well.

## What is Sentiment Analysis?

With Sentiment Analysis, we want to determine the attitude (e.g the sentiment) of for example a speaker or writer

with respect to a document, interaction, or event. Therefore it is a natural language processing problem where text needs to be understood, to predict the underlying intent. The sentiment is mostly categorized into positive, negative and neutral categories. With the use of Sentiment Analysis, we want to predict for example a customers opinion and attitude about a product based on a review he wrote about it. Because of that, Sentiment Analysis is widely applied to things like reviews, surveys, documents and much more.

The imdb Dataset

The imdb sentiment classification dataset consists of 50,000 movie reviews from imdb users that are labeled as either positive (1) or negative (0). The reviews are preprocessed and each one is encoded as a sequence of word indexes in the form of integers. The words within the reviews are indexed by their overall frequency within the dataset. For example, the integer "2" encodes the second most frequent word in the data. The 50,000 reviews are split into 25,000 for training and 25,000 for testing. The dataset was created by researchers of the Stanford University and published in a paper in 2011, where they achieved 88.89% accuracy. It was also used within the "Bag of Words Meets Bags of Popcorn" Kaggle competition in 2011.

Import Dependencies and get the Data

We start by importing the required dependencies to preprocess our data and to build our model.

```
%matplotlib inline

import matplotlib

import matplotlib.pyplot as plt

import numpy as np

from keras.utils import to_categorical

from keras import models

from keras import layers
```

We continue with downloading the imdb dataset, which is fortunately already built into Keras. Since we don't want to have a 50/50 train test split, we will immediately merge the data into data and targets after downloading, so that we can do an 80/20 split later on.

```
from keras.datasets import imdb

(training_data, training_targets), (testing_data, testing_targets) = imdb.load_data(num_words=10000)

data = np.concatenate((training_data, testing_data), axis=0)
```

targets = np.concatenate((training_targets, testing_targets), axis=0)

Exploring the Data

Now we can start exploring the dataset:

```
print("Categories:", np.unique(targets))
print("Number of unique words:", len(np.unique(np.hstack(data))))
```

```
Categories: [0 1]
Number of unique words: 9998
```

```
length = [len(i) for i in data]
print("Average Review length:", np.mean(length))
print("Standard Deviation:", round(np.std(length)))
```

```
Average Review length: 234.75892
Standard Deviation: 173.0
```

You can see in the output above that the dataset is labeled into two categories, either 0 or 1, which represents the sentiment of the review. The whole dataset contains 9998 unique words and the average review length is 234 words, with a standard deviation of 173 words.

Now we will look at a single training example:

```
print("Label:", targets[0])
```

```
Label: 1
```

```
print(data[0])
```

```
[1, 14, 22, 16, 43, 530, 973, 1622, 1385, 65, 458, 4468, 66, 3941, 4, 173, 36, 256, 5, 25, 100, 43, 838, 112, 50, 670, 2,
9, 35, 480, 284, 5, 150, 4, 172, 112, 167, 2, 336, 385, 39, 4, 172, 4536, 1111, 17, 546, 38, 13, 447, 4, 192, 50, 16, 6,
147, 2025, 19, 14, 22, 4, 1920, 4613, 469, 4, 22, 71, 87, 12, 16, 43, 530, 38, 76, 15, 13, 1247, 4, 22, 17, 515, 17, 12,
16, 626, 18, 2, 5, 62, 386, 12, 8, 316, 8, 106, 5, 4, 2223, 5244, 16, 480, 66, 3785, 33, 4, 130, 12, 16, 38, 619, 5, 25,
124, 51, 36, 135, 48, 25, 1415, 33, 6, 22, 12, 215, 28, 77, 52, 5, 14, 407, 16, 82, 2, 8, 4, 107, 117, 5952, 15, 256, 4, 2,
7, 3766, 5, 723, 36, 71, 43, 530, 476, 26, 400, 317, 46, 7, 4, 2, 1029, 13, 104, 88, 4, 381, 15, 297, 98, 32, 2071, 56, 26,
141, 6, 194, 7486, 18, 4, 226, 22, 21, 134, 476, 26, 480, 5, 144, 30, 5535, 18, 51, 36, 28, 224, 92, 25, 104, 4, 226, 65,
16, 38, 1334, 88, 12, 16, 283, 5, 16, 4472, 113, 103, 32, 15, 16, 5345, 19, 178, 32]
```

Above you see the first review of the dataset which is labeled as positive (1). The code below retrieves the dictionary mapping word indices back into the original words so that we can read them. It replaces every unknown word with a “#”. It does this by using the get_word_index() function.

```
index = imdb.get_word_index()

reverse_index = dict([(value, key) for (key, value) in index.items()])

decoded = " ".join( [reverse_index.get(i - 3, "#") for i in data[0]] )

print(decoded)
```

```
# this film was just brilliant casting location scenery story direction everyone's really suited the part they played
and you could just imagine being there robert # is an amazing actor and now the same being director # father
came from the same scottish island as myself so i loved the fact there was a real connection with this film the
witty remarks throughout the film were great it was just brilliant so much that i bought the film as soon as it was
released for # and would recommend it to everyone to watch and the fly fishing was amazing really cried at the
end it was so sad and you know what they say if you cry at a film it must have been good and this definitely was
also # to the two little boy's that played the # of norman and paul they were just brilliant children are often left
out of the # list i think because the stars that play them all grown up are such a big profile for the whole film but
these children are amazing and should be praised for what they have done don't you think the whole story was so
lovely because it was true and was someone's life after all that was shared with us all
```

Data Preparation

Now it is time to prepare our data. We will vectorize every review and fill it with zeros so that it contains exactly

10,000 numbers. That means we fill every review that is shorter than 10,000 with zeros. We do this because the biggest review is nearly that long and every input for our neural network needs to have the same size. We also transform the targets into floats.

```
def vectorize(sequences, dimension = 10000):

results = np.zeros((len(sequences), dimension))

for i, sequence in enumerate(sequences):

results[i, sequence] = 1

return results


data = vectorize(data)

targets = np.array(targets).astype("float32")
```

Now we split our data into a training and a testing set. The training set will contain 40,000 reviews and the testing set 10,000.

```
test_x = data[:10000]

test_y = targets[:10000]

train_x = data[10000:]

train_y = targets[10000:]
```

Building and Training the Model

We can now build our simple Neural Network. We start by defining the type of model we want to build. There are two types of models available in Keras: the Sequential model and the Model class used with functional API.

Then we simply add the input-, hidden- and output-layers. Between them, we are using dropout to prevent overfitting. Note that you should always use a dropout rate between 20% and 50%. At every layer, we use "Dense" which means that the units are fully connected. Within the hidden-layers, we use the relu function, because this is always a good start and yields a satisfactory result most of the time. Feel free to experiment with

other activation functions. And at the output-layer, we use the sigmoid function, which maps the values between 0 and 1. Note that we set the input-shape to 10,000 at the input-layer, because our reviews are 10,000 integers long. The input-layer takes 10,000 as input and outputs it with a shape of 50.

Lastly, we let Keras print a summary of the model we have just built.

```
# Input - Layer
model.add(layers.Dense(50, activation = "relu", input_shape=(10000, )))
# Hidden - Layers
model.add(layers.Dropout(0.3, noise_shape=None, seed=None))
model.add(layers.Dense(50, activation = "relu")
model.add(layers.Dropout(0.2, noise_shape=None, seed=None))
model.add(layers.Dense(50, activation = "relu"))
# Output- Layer
model.add(layers.Dense(1, activation = "sigmoid"))model.summary()
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 50)                500050
_________________________________________________________________
dropout_1 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_2 (Dense)              (None, 50)                2550
_________________________________________________________________
```

# Scud Arsenal

dropout_2 (Dropout) (None, 50) 0

_____

dense_3 (Dense) (None, 50) 2550

_____

dense_4 (Dense) (None, 1) 51

==========

Total params: 505,201

Trainable params: 505,201

Non-trainable params: 0

_____

Now we need to compile our model, which is nothing but configuring the model for training. We use the “adam” optimizer. The optimizer is the algorithm that changes the weights and biases during training. We also choose binary-crossentropy as loss (because we deal with binary classification) and accuracy as our evaluation metric.

```
model.compile(
optimizer = "adam",
loss = "binary_crossentropy",
metrics = ["accuracy"]
)
```

We are now able to train our model. We do this with a batch_size of 500 and only for two epochs because I recognized that the model overfits if we train it longer. The Batch size defines the number of samples that will be propagated through the network and an epoch is an iteration over the entire training data. *In general a l*arger batch-size results in faster training, but don’t always converges fast. A smaller batch-size is slower in training but it *can* converge faster. This is definitely problem dependent and you need to try out a few different values. If you start with a problem for the first time, I would you recommend to you to first use a batch-size of 32, which is the standard size.

```
results = model.fit(
train_x, train_y,
epochs= 2,
batch_size = 500,
validation_data = (test_x, test_y)
)
```

```
Train on 40000 samples, validate on 10000 samples
Epoch 1/2
40000/40000 [==============================] - 5s 129us/step - loss: 0.4051 - acc: 0.8212 - val_loss: 0.2635 - val_acc: 0.8945
Epoch 2/2
40000/40000 [==============================] - 4s 90us/step - loss: 0.2122 - acc: 0.9190 - val_loss: 0.2598 - val_acc: 0.8950
```

It is time to evaluate our model:

```
print(np.mean(results.history["val_acc"]))
```

```
0.894750000536
```

Awesome! With this simple model, we already beat the accuracy of the 2011 paper that I mentioned in the beginning. Feel free to experiment with the hyperparameters and the number of layers.You can see the code for the whole model below:

# Scud Arsenal

```
import numpy as np
from keras.utils import to_categorical
from keras import models
from keras import layers
from keras.datasets import imdb
(training_data, training_targets), (testing_data, testing_targets) = imdb.load_data(num_words=10000)
data = np.concatenate((training_data, testing_data), axis=0)
targets = np.concatenate((training_targets, testing_targets), axis=0)
def vectorize(sequences, dimension = 10000):
results = np.zeros((len(sequences), dimension))
for i, sequence in enumerate(sequences):
 results[i, sequence] = 1
return results


data = vectorize(data)
targets = np.array(targets).astype("float32")
test_x = data[:10000]
test_y = targets[:10000]
train_x = data[10000:]
train_y = targets[10000:]
model = models.Sequential()
# Input - Layer
model.add(layers.Dense(50, activation = "relu", input_shape=(10000, )))
```

# Scud Arsenal

```
# Hidden - Layers
model.add(layers.Dropout(0.3, noise_shape=None, seed=None))
model.add(layers.Dense(50, activation = "relu"))
model.add(layers.Dropout(0.2, noise_shape=None, seed=None))
model.add(layers.Dense(50, activation = "relu"))
# Output- Layer
model.add(layers.Dense(1, activation = "sigmoid"))
model.summary()
# compiling the model
model.compile(
optimizer = "adam",
loss = "binary_crossentropy",
metrics = ["accuracy"]
)
results = model.fit(
train_x, train_y,
epochs= 2,
batch_size = 500,
validation_data = (test_x, test_y)
)
print("Test-Accuracy:", np.mean(results.history["val_acc"]))
```

BY

A.N.V.Vamsi Krishna

17A81A0501

## Details of Faculty attended FDPs, Workshops, Seminars, Conference etc., outside the college as well as in the college: 06

| S.No | Name and Designation of the Faculty | Name of Workshop/Seminar/ FDP/SDP Attended | Location | No. Of Days | From Date | To Date |
|---|---|---|---|---|---|---|
| 1 | O .Sri Nagesh | Improving Teaching skills in the subject python programming | JNTUK Kakinada | 06 | 27-05-2019 | 01-06-2019 |
| 2 | M.Vamsi krishna | Improving Teaching skills in the subject python programming | JNTUK Kakinada | 06 | 27-05-2019 | 01-06-2019 |
| 3 | A leelavathi | Big Data Computing | NIT, AP | 06 | 29-05-2019 | 04-05-2019 |
| 4 | A.Rajesh | Big Data Computing | NIT, AP | 06 | 29-05-2019 | 04-05-2019 |
| 5 | N.Hiranmayi | Big Data Computing | NIT, AP | 06 | 29-05-2019 | 04-05-2019 |
| 6 | B.Sri Ramya | Big Data Computing | NIT, AP | 06 | 29-05-2019 | 04-05-2019 |

## Important Visitors to the Department: 04

| S.No | Name of Eminent Guest | Organization | Date(s) of Visit |
|---|---|---|---|
| 1. | Dr.A.Krishna Mohan | Professor, CSE, JNTUK | 20.04.2019 |
| 2. | Dr. R.B.V. Subramaanyam | Professor, CSE, NITW | 20.04.2019 |
| 3. | Dr. S. PallamSetty | Professor, CSE, AU | 20.04.2019 |
| 4. | Mr.SrinivasaRajuVuppalapati | Director, Alykas Innovations Pvt.Ltd | 20.04.2019 |
| 5. | Mr.EedalaRambabu | Alumni, Member of Technical Staff 2,Micro Focus | 20.04.2019 |

## Student Achievements

### a. Internships

| S. No | Name | RegdNo | Title | Name Of The Industry | Duration |
|---|---|---|---|---|---|
| 1 | P.Tarun Kumar | 15A81A0538 | Autocad plugin development | nCircle Tech Pvt.Ltd., A ProtoTech Solutions | 11-02-2019 To 31-05-2019 |
| 2 | Manepalli.Arunadevi | 15A81A0529 | Embedded Systems | Vaya group, Hyderabad | 3-12-2018 To 31-5-2019 |
| 3 | M.Divya Prasanna | 16A81A05K9 | Machine Learning with Python | Pivotal Soft Pvt Ltd,Visakhapatnam | 4-05-2019 To 4-06-2019 |
| 4 | S.L Keerthi | 16A81A05M5 | | | |
| 5 | V.Maha Lakshmi | 16A81A05N5 | | | |
| 6 | A. Pavani | 16A81A05I1 | | | |
| 7 | B. VenuDaneswari | 16A81A05I4 | | | |
| 8 | P.Madhu Annapurna | 16A81A05L8 | | | |
| 9 | V.C.V.L Mounika | 16A81A05N3 | | | |
| 10 | V.Navya Srilatha | 16A81A05N4 | | | |
| 11 | Shaik Sabeen | 16A81A05M9 | | | |
| 12 | V. Rama Leela Sai | 16A81A05N6 | | | |
| 13 | N. Devi kataksham | 16A81A05L0 | | | |
| 14 | G.Roja Rani | 17A85A0502 | IOT | ORL industries | 13-05-2019 To 15-06-2019 |
| 15 | P. Valli | 17A85A0504 | | | |
| 16 | P. Sravani | 16A81A05L9 | | | |
| 17 | K.Krishna Vamsi Sai | 16A81A05J6 | Web Design | Kate Technologies,Hyd | 16-05-2019 T0 18-06-2019 |
| 18 | A.Manikanta | 16MU1A0525 | Android | Pivotal Soft Pvt Ltd,Visakhapatnam | 4-05-2019 To 4-06-2019 |
| 19 | K.Satya Sri | 16A81A05J7 | Web Design | Miracle Software Systems | 13-05-2019 To 14-06-2019 |
| 20 | S. Harshini Sai | 16A81A05M4 | IOT | Mad blocks Technologies Pvt Ltd | 13-05-2019 To 13-06-2019 |
| 21 | K.P.S Sireesha | 17A85A0505 | | | |
| 22 | T.Karishma Reddy | 16A81A05N1 | | | |
| 23 | Y . Mounika Lakshmi | 16A81A0559 | IOT & Robotics | | |

| | | | | | |
|---|---|---|---|---|---|
| 24 | P. Kanyaka Alekya | 17A81A0594 | Recruitment Interview Scheduling App | Miracle Software Systems | 13-05-2019 To 14-06-2019 |
| 25 | E. Hema Latha | 16A81A05D2 | Conference Room Booking and Status App | | |
| 26 | Amisetti Vamsi Krishna | 17A81A0501 | Marketing | MYCAPTAIN | 25-10-2018 to 31-11-2018 |

**b.The following list of students completed the APSSDC –Gamification workshop , from 16.05.2019 to 31.05.2019.**

| S.No. | Reg. Number | Name of the Student |
|---|---|---|
| 1 | 17A81A0501 | A N V Vamsi Krishna |
| 2 | 17A81A0503 | BALAGAM SWATHI |
| 3 | 17A81A0513 | GANGULA VENKATA LAKSHMI TULASI |
| 4 | 17A81A0524 | KUCHIBHOTLA SRI SATYA SRAVYA |
| 5 | 17A81A0543 | SIMHADRI TEJA SRI |
| 6 | 17A81A0557 | BALUSU LAVANYA |
| 7 | 17A81A0566 | GODAVARTHI BHARGAVI |
| 8 | 17A81A0572 | JUVVALAPALEPU PRAMEELA |
| 9 | 17A81A0579 | KUNIREDDY DEVI SAI DURGA |
| 10 | 17A81A0590 | PANDIRIPALLI MANJUSHA |
| 11 | 17A81A0594 | PEDAMALLU KANYAKA ALEKYA |
| 12 | 17A81A05B3 | AdabalaSatish |
| 13 | 17A81A05C4 | Chukka Shyam Kumar |
| 14 | 17A81A05E5 | N Teja Kiran |
| 15 | 17A81A05F4 | R Sai Kiran |
| 16 | 17A81A05F6 | B Utthej Kumar |
| 17 | 17A81A05F9 | M V D S Vardhan |
| 18 | 17A81A05I7 | KANCHERLA SAHITHI CHANDRA TEJASWINI |
| 19 | 17A81A05L4 | VEMULA PAVANI SIVA PRATHYUSHA |
| 20 | 17A81A05L5 | VISHNUMOLAKALA BADARINADH |
| 21 | 17A81A05K4 | PULLURI ADILAKSHMI |
| 22 | 17A81A05H8 | CHITTURI LATHA SRAVANI |
| 23 | 17A81A05H3 | BHOGARAJU L N LALITHA PRABHA |
| 24 | 17A81A05G9 | ATYAM N V S S S L RAMYA |

| | | |
|---|---|---|
| 25 | 17A81A05I5 | JAKKI PRAGATHISREE |
| 26 | 17A81A05H6 | CHALLA VENKATA SAI RAM |
| 27 | 17A81A0516 | GOTETI B S Y VASUDEVA RAO |
| 28 | 17A81A05H4 | BOLLAM RENUKA VENKATA PRIYANKA |

**c. NPTEL Certified Students List (Jan-April-2019 Session)**

| S.No. | Roll Number | Name of the student | Course Name | Certificate Type |
|---|---|---|---|---|
| 1 | 17A81A0595 | BhavaniPosimsetty | Programming in Java | Successfully completed |
| 2 | 17A81A0598 | Reddy Akanksha | | Elite+Silver |
| 3 | 17A81A0594 | PedamalluKanyakaAlekya | | Successfully completed |
| 4 | 17A81A05F4 | RayabarapuSaikiran | | Elite+Silver |
| 5 | 17A81A05G7 | Sri BhavaniYallavula | | Elite+gold |
| 6 | 17A81A05C5 | DamisettiDeepika Naga Ratnam | | Elite+gold |
| 7 | 17A81A05I4 | ChaitanyaIndugula | Programming, Data Structures and Algorithms using Python | Elite |
| 8 | 16A81A05F4 | Kommuri Naga SrideviDivya | Data Mining | Elite |
| 9 | 16A81A0543 | SnehithaPenugonda | | Successfully completed |
| 10 | 16A81A0534 | MandapakaPavan Kumar | | Elite |
| 11 | 16A81A0539 | P Salma Begum | | Successfully completed |
| 12 | 16A81A0533 | Surendra Reddy | | Successfully completed |
| 13 | 16A81A0555 | SwethaKiranmayi | | Successfully completed |
| 14 | 16A81A0520 | ViswaSaiKarpurapu | | Successfully completed |
| 15 | 16A81A05C6 | B.V.L.R.Anjali | | Elite |
| 16 | 16A81A05C2 | B. Devi Surekha | | Elite |
| 17 | 16A81A05F3 | K.Sushmakumari | | Successfully completed |
| 18 | 17A81A0501 | Amisetti N V Vamsi Krishna | Introduction to Internet of Things | Elite |

| | | | | |
|---|---|---|---|---|
| 19 | 18A81D5302 | PaletiJyothsnaSree | Big Data Computing | Successfully completed |
| 20 | 18A81D5301 | ChippadaNarayanaBhargav | | Successfully completed |
| 21 | 18A81D5303 | PaturiManikantha | | Successfully completed |

## NPTEL Certified Faculty List (Jan-April-2019 Session)

| S.No. | Name of the Faculty | Course | Certificate Type |
|---|---|---|---|
| 1. | AnjaniSuputri Devi D | Compiler Design | Elite |
| 2. | Praveen Kumar Nallabariki | | Successfully Completed |
| 3. | R.Vinupriya | Computer Architecture And Organisation | Elite |
| 4. | KoradaRajaniKumar | | Successfully Completed |
| 5. | LeelavathiArepalli | Problem Solving Through Programming In C | Elite |
| 6. | MademNageswara Rao | | Elite |
| 7. | V K Hanuman Turaga | Programming In Java | Elite+Silver |
| 8. | Ch Raja Ramesh | | Elite+Silver |
| 9. | R Leele Phani Kumar | | Elite+Silver |
| 10. | Shirin Bhanu Koduri | | Elite+Gold |
| 11. | Nataraj Gudapaty | | Elite |
| 12. | Praneetha Pitta | | Elite+Silver |
| 13. | Rama Rajeswari Mulukutla | | Successfully Completed |
| 14. | VrpsSastryYadavilli | Programming, Data Structures And Algorithms Using Python | Successfully Completed |
| 15. | LakshminarayanaKondreddi | | Elite |
| 16. | SdSabirHussain | | Successfully Completed |
| 17. | A.Sirisha | Programming In C++ | Successfully Completed |
| 18. | M R Raja Ramesh | Data Base Management System | Elite+Silver |
| 19. | Sukanya | | Elite+Silver |
| 20. | ItteLavanya | | Elite+Silver |
| 21. | Dr.D Jaya Kumari | Data Mining | Elite |
| 22. | Loshma Gunisetti | | Elite+Silver |
| 23. | AlamuruVenkannababu | | Elite |
| 24. | Sp.Malarvizhi | | Elite+Silver |
| 25. | PusuluruHariChandana | | Elite |
| 26. | D. SasiRekha | Real Time Operating System | Successfully Completed |

| | | | |
|---|---|---|---|
| **27.** | Ch Raja Ramesh | Machine Learning | Successfully Completed |
| **28.** | KavidiVenkata Mutyalu | | Successfully Completed |
| **29.** | KavidiVenkata Mutyalu | Social Networks | Successfully Completed |
| **30.** | GokavarapuSriram Ganesh | Introduction To Automata, Languages And Computation | Elite+Silver |

## Sayonara-2k19 conducted on 04.04.2019

### Placements

| *Roll No.* | *Name of the Student* | *Company* | *CTC (LPA)* |
|---|---|---|---|
| **15A81A0501** | ALLA LAKSHMI BHARGAVI | **TCS CODE VITA** | **3.6 LPA** |
| **15A81A0528** | MAJETY H V A SRI VYSHNAVI | | |
| **15A81A0536** | NEELAM PRUDHVINI SAI | | |
| **15A81A0546** | THOTA NEERAJA | | |
| **15A81A05C0** | VUDDAGIRI INDUMATHI | | |
| **15A81A05F1** | MADDAMSETTI LAKSHMI MOUNICA | | |
| **15A81A0514** | KEDARISETTI DEEPIKA | **MPHASIS** | **1.8 LPA** |
| **15A81A0517** | KALLA SATYAVATHI | | |
| **15A81A0536** | NEELAM PRUDHVINI SAI | | |
| **15A81A0546** | THOTA NEERAJA | | |
| **15A81A0562** | AINAMPUDI LAVANYA DEVI | | |
| **15A81A0576** | KANDUKURI AKHILA | | |
| **15A81A0588** | MAMIDALA PURNA VAMSI | | |
| **15A81A0595** | DURGAPRIYA NOWBATHULA | | |
| **15A81A0597** | PAILU KAVYA PRIYA | | |
| **15A81A05B4** | SUREDDY KRISHNA SAI PHANI KUMAR | | |
| **15A81A05B9** | VANDANAPU LASYAPRIYA | | |
| **15A81A05C4** | ALLAM SNEHA SRI SANDHYA | | |
| **15A81A05F2** | VARALAKSHMI MAHANTHI | | |
| **15A81A05G8** | SARIDE KARTHIK | | |
| **15A81A05H0** | SEELAM SOWJANYA | | |
| **15A81A0515** | KALAVALAPALLI BHASKAR GANESH | **L&T TECHNOLOGY** | **4.0 LPA** |

# Scud Arsenal

| | | | |
|---|---|---|---|
| **15A81A0526** | KOTHA SWATHI SRI SAI NAGA DURGA | **SERVICES** | |
| **15A81A0552** | VARANASI LAKSHMIMEGHANA | | |
| **15A81A0555** | VELIGATLA SHAMITHA | | |
| **15A81A0565** | ATMAKURI SAI RUCHITHA | | |
| **15A81A0573** | GONTLA LAKSHMI INDIRA | | |
| **15A81A05A2** | PENDYALA DURGA SUNEETHA | | |
| **15A81A05A6** | SAMAYAMANTHULA J LAKSHMI SRI | | |
| **15A81A05C0** | VUDDAGIRI INDUMATHI | | |
| **15A81A05D1** | CHILUKURI LAKSHMI CHANDINI | | |
| **15A81A05D9** | GRANDHI PRAGATHI RAO | | |
| **15A81A05F1** | MADDAMSETTI LAKSHMI MOUNICA | | |
| **15A81A05F5** | MUSUNURU BALA KRISHNA | | |
| **15A81A05J5** | DONE RESHMA DHAR | | |
| **15A81A05K0** | KARRI CHIRANJEEVI SAI ISWARYA | | |
| **15A81A05L3** | MANGENA GIRI NAGA PAVAN KUMAR | | |
| **15A81A05M6** | POSINA SASANKA | | |
| **15A81A05M8** | SAJJA SAI KUMAR | | |
| **16A85A0503** | KAVALA SAI RAMESH | | |
| **15A81A0510** | GARLAPATI NAVYA | **TCS NINJA** | **3.5 LPA** |
| **15A81A0547** | TUMMALAPALLI SUPRIYA | | |
| **15A81A0554** | V.VINEELA | | |
| **15A81A0555** | VELIGATLA SHAMITHA | | |
| **15A81A0565** | ATMAKURI SAI RUCHITHA | | |
| **15A81A0595** | DURGAPRIYA NOWBATHULA | | |
| **15A81A0596** | PADMANABHUNI MANIDEEP | | |
| **15A81A0597** | PAILU KAVYA PRIYA | | |
| **15A81A05A1** | PATCHIPULUSU A L SOWMYA SRI | | |
| **15A81A05D7** | G.HEMALATHA | | |
| **15A81A05G2** | YAMUNA PANCHADI | | |
| **15A81A05I0** | Y.BHANUPRIYA | | |
| **15A81A05K9** | KUSAM VENKATA ALEKHYA | | |
| **15A81A05M0** | NARRA MOUNIKA | | |
| **15A81A0525** | KOTHA NALINI VENKATA LAKSHMI | **EFFTRONICS SYSTEMS PVT. LTD.,** | **3 - 7 LPA** |
| **15A81A0538** | TARUN KUMAR POLAVARAPU | **PROTOTECH SOLUTIONS** | **2.80 LPA** |

# Scud Arsenal

| | | | |
|---|---|---|---|
| **15A81A0518** | KANDRU NAGA MOUNIKA | **OPSRAMP PVT. LTD.,** | **3.3 LPA** |
| **15A81A0550** | VANNEMREDDY RAMA LAKSHMI | | |
| **15A81A0578** | KATHARI SAI JOSHITHA | | |
| **15A81A0580** | KODE SUBHASHINI | | |
| **15A81A05F8** | NEKKANTI VYSHNAVI | | |
| **15A81A05K1** | KATTA VASAVI VIDYA SIREESHA | | |
| **15A81A0586** | MADUPALLI JAYANTH | **SNOVASYS SOFTWARE SOLUTIONS PVT. LTD.,** | **3.4 LPA** |
| **15A81A0506** | CHALLA BALA SAI SRILEKHA | **SYNTEL** | **3.0 LPA** |
| **15A81A0510** | GARLAPATI NAVYA | | |
| **15A81A0518** | KANDRU NAGA MOUNIKA | | |
| **15A81A0519** | KANKATALA LAKSHMI VINEELA | | |
| **15A81A0527** | L INDIRA PRAVALLIKA | | |
| **15A81A0529** | MANEPALLI ANURUPADEVI | | |
| **15A81A0531** | LIKHITHA MATHI | | |
| **15A81A0545** | TELAGAMSETTI SUPRIYA | | |
| **15A81A0558** | VIPPARTHI N K S MANJU SREE | | |
| **15A81A0569** | DAGGU VENKATA NAVEEN | | |
| **15A81A0573** | GONTLA LAKSHMI INDIRA | | |
| **15A81A0578** | KATARI SAI JOSHITHA | | |
| **15A81A0580** | KODE SUBHASHINI | | |
| **15A81A0599** | PALURI SRI HARIKA | | |
| **15A81A05A6** | JHANSI L S SAMAYAMANTHULA | | |
| **15A81A05B1** | SRIPADA DIVYA SATYA SAI | | |
| **15A81A05C3** | ALLA NAGAVALLI DEVI | | |
| **15A81A05C9** | BUDDANA SRAVYA TULASI | | |
| **15A81A05F8** | NEKKANTI VYSHNAVI | | |
| **15A81A05G8** | SARIDE KARTHIK | | |
| **15A81A05I1** | ADDANKI DIVYA | | |
| **15A81A05K0** | KARRI CHIRANJEEVI SAI ISWARYA | | |
| **16A85A0508** | TAMMISETTI VIJAY SUNDAR | | |
| **15A81A0529** | MANEPALLI ANURUPADEVI | **VAYA GROUP** | **2.4 LPA** |
| **15A81A0550** | VANNEMREDDY RAMA LAKSHMI | | |
| **15A81A05K1** | KATTA VASAVI VIDYA SIREESHA | | |
| **15A81A0505** | CHALAMALASETTI JHANDINI | **WIPRO** | **3.50 LPA** |

# Scud Arsenal

| | | | |
|---|---|---|---|
| **15A81A0528** | MAJETY H V A SRI VYSHNAVI | | |
| **15A81A0536** | NEELAM PRUDHVINI SAI | | |
| **15A81A0540** | RALLAPALLI MOUNICA | | |
| **15A81A0547** | TUMMALAPALLI SUPRIYA | | |
| **15A81A0552** | VARANASI LAKSHMIMEGHANA | | |
| **15A81A0562** | AINAMPUDI LAVANYA DEVI | | |
| **15A81A0563** | ALETI J VENKATA GOPI MANIKANTA | | |
| **15A81A0576** | K.AKHILA | | |
| **15A81A0580** | KODE SUBHASHINI | | |
| **15A81A0581** | KODI JYOTHIRMAI | | |
| **15A81A0595** | DURGAPRIYA NOWBATHULA | | |
| **15A81A05A2** | PENDYALA DURGA SUNEETHA | | |
| **15A81A05B4** | SUREDDY KRISHNA SAI PHANI KUMAR | | |
| **15A81A05B6** | THANMAI SATYA SAI SRI LAKSHMI T | | |
| **15A81A05C0** | VUDDAGIRI INDUMATHI | | |
| **15A81A05F1** | MADDAMSETTI LAKSHMI MOUNICA | | |
| **15A81A05G2** | PANCHADI YAMUNA | | |
| **15A81A05K0** | KARRI CHIRANJEEVI SAI ISWARYA | | |
| **15A81A05K1** | KATTA VASAVI VIDYA SIREESHA | | |
| **15A81A05K6** | KORLEPARA L R SAI SWATHI SRI | | |
| **15A81A05K8** | KURALLA NAGA LAKSHMI SUSMITHA | | |
| **15A81A0505** | CHALAMALASETTI JHANDINI | **MAGNAQUEST** | **3.0 LPA** |
| **15A81A05E6** | KORLEPARA SAI LAKSHMI PRASANNA | | |
| **15A81A0525** | KOTHA NALINI VENKATA LAKSHMI | **ZEN TECHNOLOGIES** | **3.5 - 5.00 LPA** |
| **15A81A0525** | KOTHA NALINI VENKATA LAKSHMI | **INFOSYS** | **3.60 LPA** |
| **15A81A0580** | KODE SUBHASHINI | | |
| **15A81A05C5** | ALLU RANI SRINIJA | | |
| **15A81A05E1** | GULLAPUDI DHATRI SRI VINAYA | | |
| **15A81A05F8** | NEKKANTI VYSHNAVI | | |
| **15A81A05I1** | ADDANKI DIVYA | | |
| **15A81A0563** | ALETI J VENKATA GOPI MANIKANTA | **CADDYCODE** | **2.40 LPA** |
| **15A81A0594** | NANDURI NAGA VENKATA JAHNAVI | | |
| **15A81A05K7** | KOTARU SOWJANYA | | |
| **15A81A0509** | EDADASULA PARVATHI | **CGI** | **3.30 LPA** |
| **15A81A0530** | MANGENA LAKSHMI SUPRAJA | | |

# Scud Arsenal

| | | | |
|---|---|---|---|
| **15A81A0543** | TAMMISETTI VIJAYALAKSHMI | | |
| **15A81A05E6** | KORLEPARA SAI LAKSHMI PRASANNA | | |
| **15A81A0523** | KOMPELLA NAGASATYA PRAVALLIKA RAMYASRI | **COGNIZANT** | **3.38 LPA** |
| **15A81A0532** | MEESALA BALA DURGA | | |
| **15A81A0543** | TAMMISETTI VIJAYALAKSHMI | | |
| **15A81A0575** | MRUDHULA SRI KAMMA | | |
| **15A81A0579** | KIRALA RISHIKA RANI | | |
| **15A81A0589** | MARAGANI SRAVANI DURGA | | |
| **15A81A05C8** | BOMMA JAGRUTHI S V P DURGAMANI | | |
| **15A81A05D4** | DAMARAJU N S K SINDHU PRIYA | | |
| **15A81A05H3** | TEKI KALYANI | | |
| **15A81A05J0** | CHELLANKI RAMYA SUDHA | | |
| **15A81A05L4** | MOHAMMAD YASMIN | | |
| **15A81A05D4** | DAMARAJU N S K SINDHU PRIYA | **PENNANT TECHNOLOGIES** | **1.80 - 2.40 LPA** |
| **15A81A05H1** | SHAIK IZAZ AHMAD | | |
| **15A81A05J4** | CHUNDURU NAGA PRATYUSHA | | |
| **15A81A0550** | VANNEMREDDY RAMA LAKSHMI | **VIRTUSA POLARIS** | **3.30 LPA** |
| **15A81A0575** | MRUDHULA SRI KAMMA | | |
| **15A81A05F1** | MADDAMSETTI LAKSHMI MOUNICA | | |
| **15A81A05F2** | MAHANTHI VARA LAKSHMI | | |
| **15A81A0541** | SAREDDY SUPRIYA | **MIRACLE SOFTWARE SYSTEMS** | **1.80 LPA** |
| **15A81A0564** | ANKAM MAHESH KUMAR | | |
| **15A81A0579** | KIRALA RISHIKARANI | | |
| **15A81A0587** | MALLABATHULA HEMA MOUNIKA | | |
| **15A81A0589** | MARAGANI SRAVANI DURGA | | |
| **15A81A05B2** | SRIRAM VENKATA SUBBA RAO | | |
| **15A81A05B5** | TADURI SREERAMANI CHANDRIKA | | |
| **15A81A05C6** | ANNAMREDDY YAGNA PRIYA KUMAR | | |
| **15A81A05J0** | CHELLANKI RAMYA SUDHA | | |
| **15A81A05K3** | KODUGANTI SREE SATYA KANAKAVALLI | | |
| **15A81A05L6** | MUTYAM VASANTHALAKSHMI | | |
| **15A81A05N4** | TAMMINA SAI DEEPIKA | | |
| **16A85A0505** | KOSURI SUVARNA RAJU | | |
| **15A81A05C2** | AKULA ANU | **TECH MAHINDRA** | **1.80 LPA** |
| **15A81A0532** | MEESALA BALA DURGA | **SYNTEL** | **3.0 LPA** |

# Scud Arsenal

| | | | |
|---|---|---|---|
| **15A81A0541** | SAREDDY SUPRIYA | | |
| **15A81A0575** | KAMMA MRUDHULA SRI | | |
| **15A81A0589** | MARAGANI SRAVANI DURGA | | |
| **15A81A0594** | NANDURI NAGA VENKATA JAHNAVI | | |
| **15A81A05A9** | SATTI DURGA | | |
| **15A81A05B3** | SURAMPUDI VINAY | | |
| **15A81A05B8** | VALLIPALLI HANUMA NAGA SUSMITHA | | |
| **15A81A05C8** | BOMMA JAGRUTHI S V P DURGA MANI | | |
| **15A81A05E8** | KUNUKU VENKATA SASI KUMAR | | |
| **15A81A05F6** | NADIMPALLI HEMA LAKSHMI BHAVANI | | |
| **15A81A05G0** | NOWDU MADHURI BHAVANI | | |
| **15A81A05J6** | ELLURI HARI BABU | | |
| **15A81A05K7** | KOTARU SOWJANYA | | |
| **15A81A05N2** | SUGGISETTI SOWMYA SRI | | |
| **15A81A05N8** | VASA CHAITANYA | | |
| **15A81A05B2** | SRIRAM VENKATA SUBBA RAO | **MPHASIS** | **1.80 LPA** |
| **15A81A05L0** | MADA SUPRAGNA | | |
| **15A81A05K8** | KURALLA NAGA LAKSHMI SUSMITHA | **MINDTREE** | **3.30 LPA** |
| **15A81A0524** | KONDAPALLI SIVA RAMA KRISHNA | **KARVY** | **1.80 LPA** |
| **15A81A0561** | A BALA VENKATA KIRAN TEJA | | |
| **15A81A0568** | CHITYALA SRI ABHIRAM | | |
| **15A81A0570** | DASI JOHN RATNA PAUL | | |
| **15A81A05A0** | PANCHAKARLA VIJAY KUMAR | | |
| **15A81A05C1** | ADAPA SARATH KUMAR | | |
| **15A81A05D0** | CHEERA GANESH | | |
| **15A81A05G7** | SABBELLA VENKATA REDDY | | |
| **15A81A05H9** | YEGIREDDY SAI PHANINDRA KUMAR | | |
| **15A81A0533** | MUNGARA NAVEEN KUMAR | **JUST DIAL** | **2.04LPA** |
| **15A81A05L2** | MALLIPUDI LOHITH MANIKANTA SANTHOSH | | |
| **16A85A0505** | KOSURI SUVARNA RAJU | | |

# Scud Arsenal

## SNIPPETS

**WHAT IS THE OUTPUT OF FOLLOWING :**

1.
```
#include<stdio.h>
int main()
{
    signed  char chr=128;
    printf("%d\n",chr);
    return 0;
}
```

a)128

b)-128

c)based on compiler

d)None

2.
```
#include<stdio.h>
int main()
{
        short  i;
        for(i=1;i>=0;i++)
        printf("%d\n",chr);
}
```

a. control won't fall into loop

b. numbers will be displayed until signed limit and throw runtime error

c. numbers will be displayed until signed limit and program terminates

d. it will be infinite loop printing numbers without error

3.
```
#include <stdio.h>
void main()
{
int k = 0;
for (k)
printf("Hello");
}
```

a) Compile time error

b) hello

c) Nothing

d) Varies

4.
```
#include <stdio.h>
void main()
{
int k = 0;
for (k < 3; k++)
printf("Hello");
}
```

a) Compile time error

b) Hello is printed thrice

c) Nothing

d) Varies

5.
```
#include <stdio.h>
void main()
{
double k = 0;
for (k = 0.0; k < 3.0; k++)
printf("Hello");
}
```

a) Run time error

b) Hello is printed thrice

c) Hello is printed twice

d) Hello is printed infinitely

# Scud Arsenal

6.
```
#include <stdio.h>
int main()
{
int a = 1, b = 1, c;
c = a++ + b;
printf("%d, %d", a, b);
}
```

a) a = 1, b = 1

b) a = 2, b = 1

c) a = 1, b = 2

d) a = 2, b = 2

7.
```
#include <stdio.h>
int main()
{
int a = 10, b = 10;
if (a = 5)
b--;
printf("%d, %d", a, b--);
}
```

a)a = 10, b = 9

b) a = 10, b = 8

c) a = 5, b = 9

d) a = 5, b = 8

8.
```
#include <stdio.h>
int main()
{
int i = 0;
int j = i++ + i;
printf("%d\n", j);
}
```

a) 0

b) 1

c) 2

d) Compile time error

ANSWERS: 1-b , 2-c , 3-a , 4-a , 5-b , 6-b , 7-c , 8-b

z

You can also send your articles for future issues to mail id:

cseskud@gmail.com