

LAB 01

21CSE295T

Statistical Data Analytics And Integration

Unit 1 Introduction to Statistical Analysis

#	LAB TASK
1	Create a SAS Profile in website (Signup)
2	Login and Launch SAS Studio
3	Write Your First SAS Program
4	Understand Your First SAS Program
5	Understand basics of SAS Program
6	SAS Program – Simple Arithmetic
7	Simple Arithmetic – Play with it!

Virupakshan K

+91-9840847872 | VIRU@KART.LA

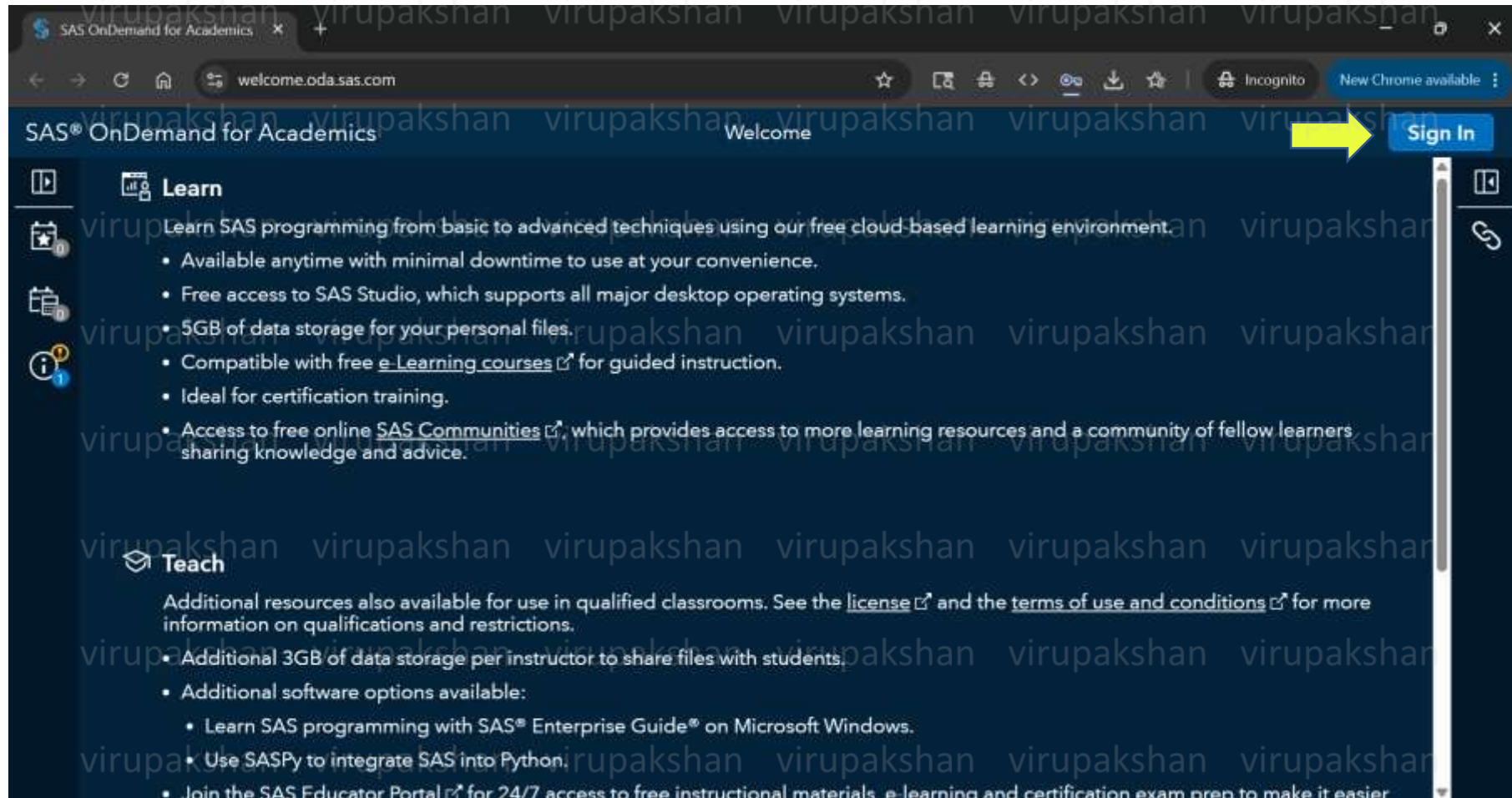
Tutorial Syllabus

1.12	SAS Statements
1.13	Structure
1.14	Libraries
1.15	Rules
1.16	Datasets
1.17	Representation of Missing Values
1.18	Assigning Libraries
1.19	PROC CONTENTS
1.20	Visualizing Descriptors and Dataset Attributes

LAB TASK 01: CREATE A SAS Profile



Step 1 : Goto <https://welcome.oda.sas.com/>



Step 2: Click on Sign In

SAS OnDemand for Academics

Welcome

Sign In

SAS Profile email address or user ID

Password

☐ Accept the terms of the license, the terms of use and conditions, and the use of cookies.

Sign In

[Forgot Password?](#)

[Don't have a SAS Profile?](#)

[Sign In not working?](#)

Learn

Learn SAS programming from basic to advanced. Available anytime with minimal download.

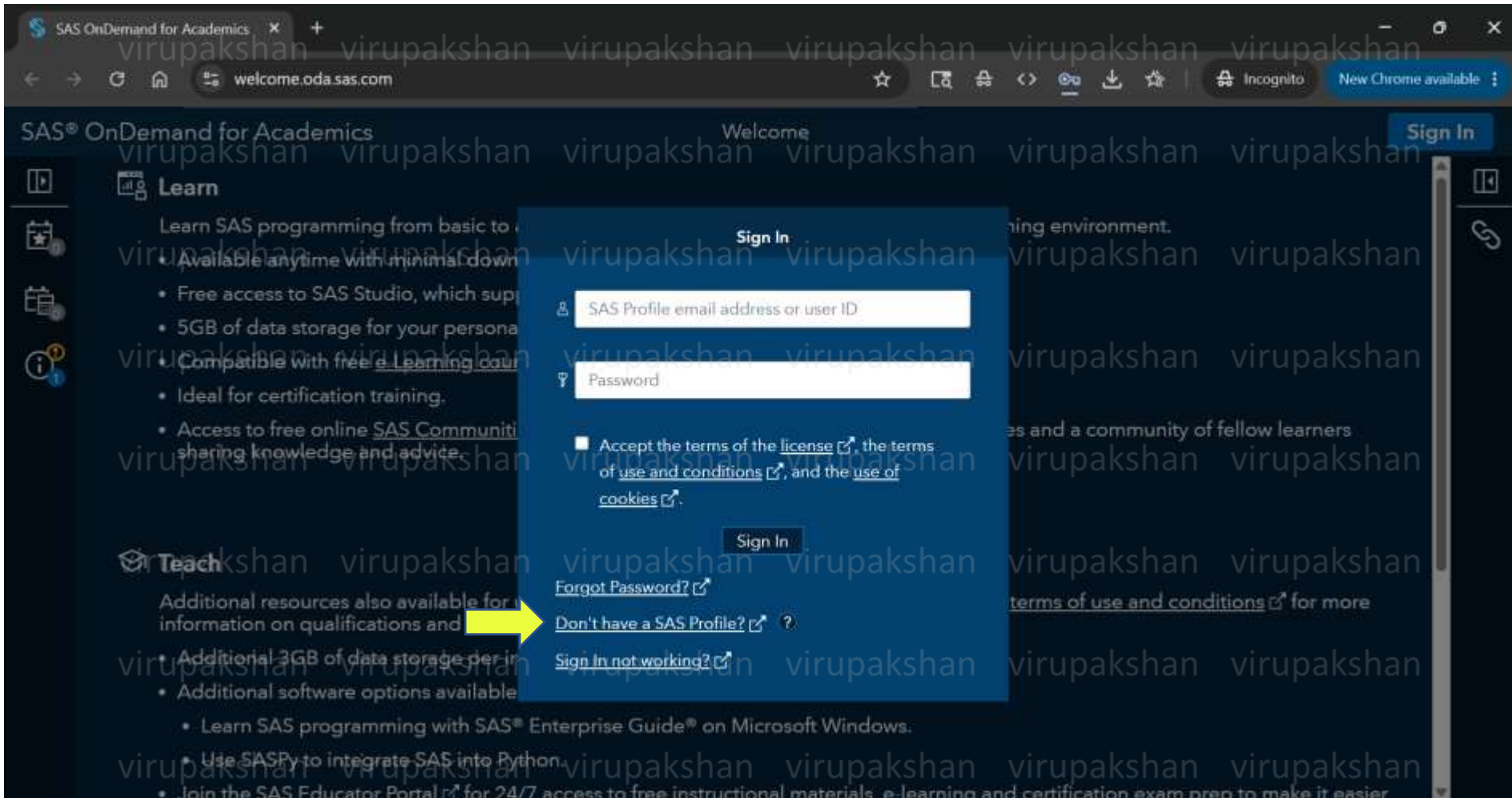
- Free access to SAS Studio, which supports SAS programming, data management, and reporting.
- 5GB of data storage for your personal projects.
- Compatible with free e-Learning courses.
- Ideal for certification training.
- Access to free online SAS Community for sharing knowledge and advice.

Teach

Additional resources also available for users with specific qualifications and restrictions.

- Additional 3GB of data storage per user.
- Additional software options available.
- Learn SAS programming with SAS® Enterprise Guide® on Microsoft Windows.
- Use SASPy to integrate SAS into Python.
- Join the SAS Educator Portal for 24/7 access to free instructional materials, e-learning and certification exam prep to make it easier.

Step 3: Click on “Don’t have a SAS Profile?” link



Step 3: Create a profile <https://www.sas.com/profile/ui/#/create>

The screenshot shows a web browser window with the URL `sas.com/profile/ui/#/create`. The page is titled "SAS Profile" and features a yellow arrow pointing to the "Preferred language" dropdown menu. The form includes the following fields:

- Preferred language (dropdown menu)
- First Name (text input)
- Last Name (text input)
- Email (text input)
- Password (text input)
- Confirm Password (text input)
- Country (dropdown menu)
- State/Province/Region (dropdown menu)
- City (text input)
- Department (text input)
- Organization (text input)
- Phone (text input)

At the bottom of the form, there is a "Create profile" button. Below the button, there is a disclaimer: "After clicking 'Create profile' you will receive a confirmation email with instructions for setting your password and activating your profile."

Step 3b: Create a profile <https://www.sas.com/profile/ui/#/create>

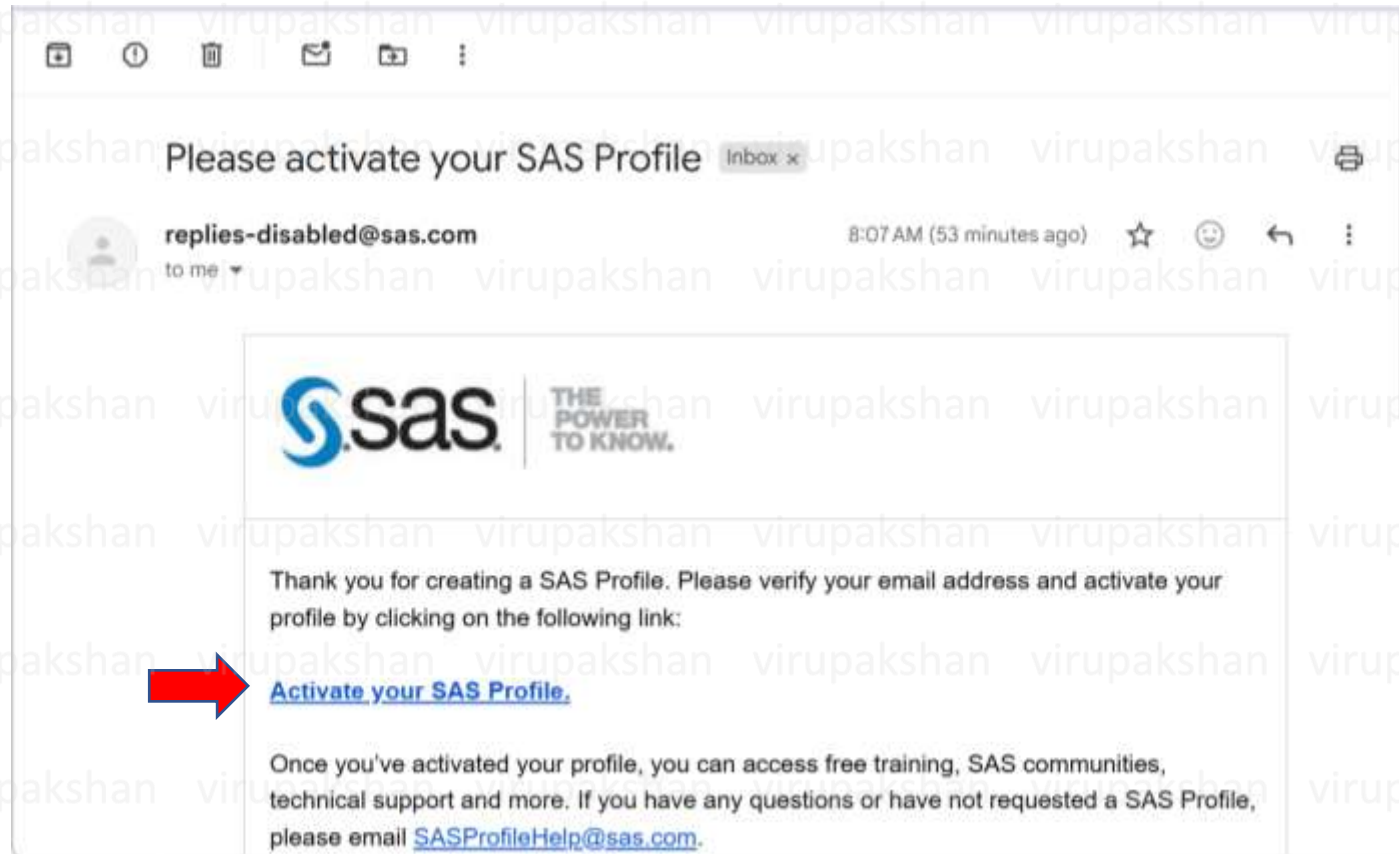
VirupakshanK@gmail.com

The screenshot displays the SAS Profile creation interface. The browser address bar shows `sas.com/profile/ui/#/create`. The page title is "SAS Profile" with the subtitle "Step 1 of 2: Tell us about yourself." The form includes the following fields:

- Preferred Language: English
- First Name: Virupakshan
- Last Name: Krishnamoorthi
- Email: virupakk@srmist.com
- Country/Region: India
- State/Province/Territory: Tamil Nadu
- Affiliation With SAS: Student
- SAS Expertise Level: Novice
- Organization/University: SRMIST
- Phone: 9962477866

At the bottom, there are checkboxes for "Yes, I would like to receive occasional emails from SAS Institute..." and "Secure my account with two-factor authentication...". A checkbox for "I agree to the terms of use and conditions" is checked. A "Create profile" button is located at the bottom right. A yellow arrow points to the "Create profile" button in the bottom left corner of the form.

Step 4: Activate your account (Check your email Inbox)



LAB TASK 02: Login and Launch SAS Studio



L02.1 Open SAS Studio

1. Login to SAS Studio ↗ <https://welcome.oda.sas.com/>
2. After login, you will see:
 - Left panel → Files/Folders
 - Center area → Code Editor
 - Bottom → Log and Results

The three most important parts are:

Area	Purpose
Code Editor	Write SAS programs
Log	Shows errors/warnings
Results	Shows output tables/charts

L02.2 Login to SAS

<https://welcome.oda.sas.com/>

SAS® OnDemand for Academics

Welcome

Sign In

You have successfully signed out of SAS OnDemand for Academics. To ensure the security of your account, please close all browser windows.

Sign In

virupakk

.....

☐ Accept the terms of the [license](#), the terms of [use and conditions](#), and the [use of cookies](#).

Sign In

[Forgot Password?](#)

[Don't have a SAS Profile?](#)

[Sign In not working?](#)

[terms of use and conditions](#) for more

Learn

Learn SAS programming from basic to advanced.

- Available anytime with minimal download.
- Free access to SAS Studio, which supports SAS programming, data management, and visualization.
- 5GB of data storage for your personal use.
- Compatible with free e-Learning courses.
- Ideal for certification training.
- Access to free online SAS Community sharing knowledge and advice.

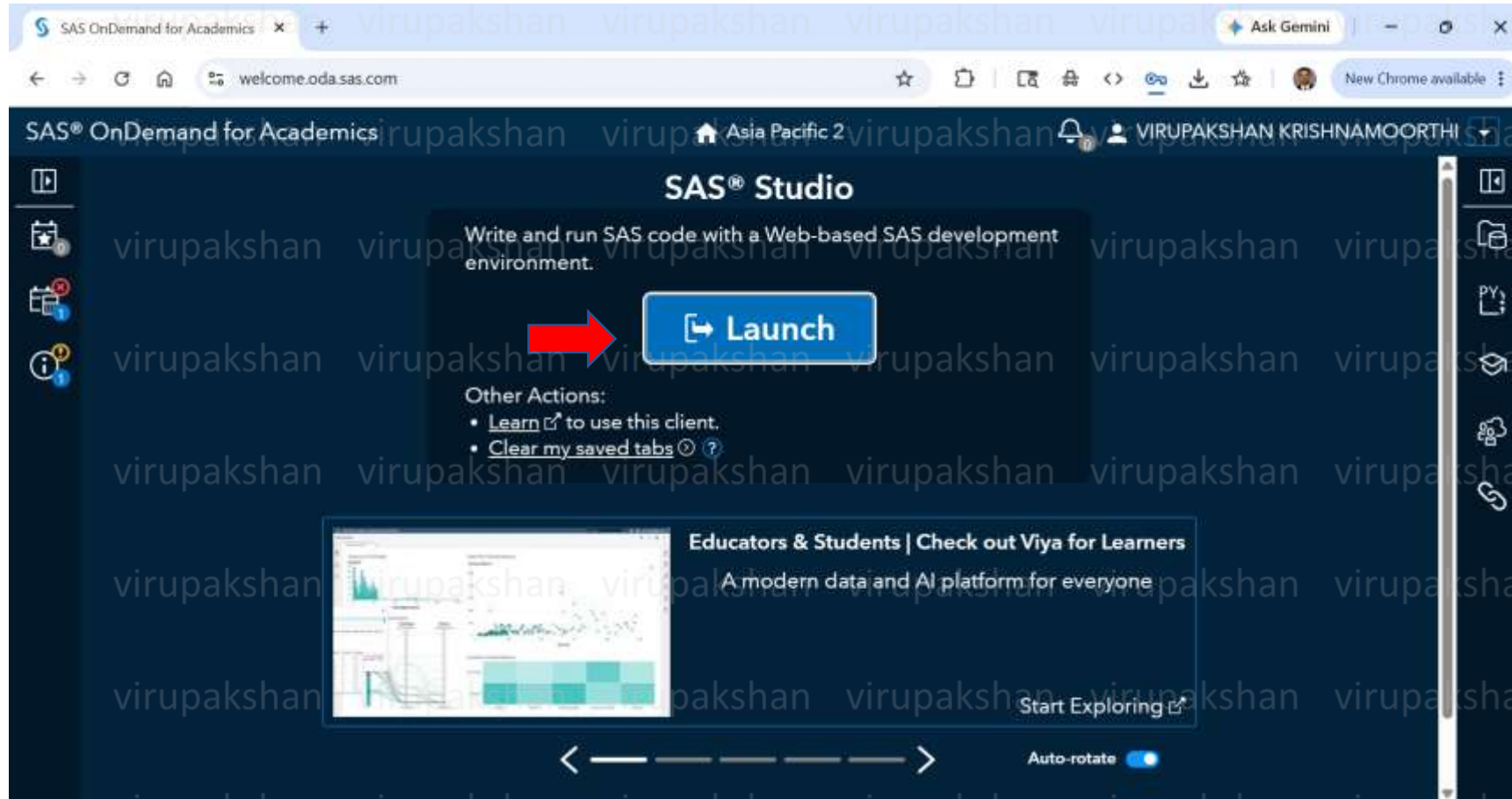
Teach

Additional resources also available for instructors.

- Additional 3GB of data storage per instructor to share files with students.
- Additional software options available:

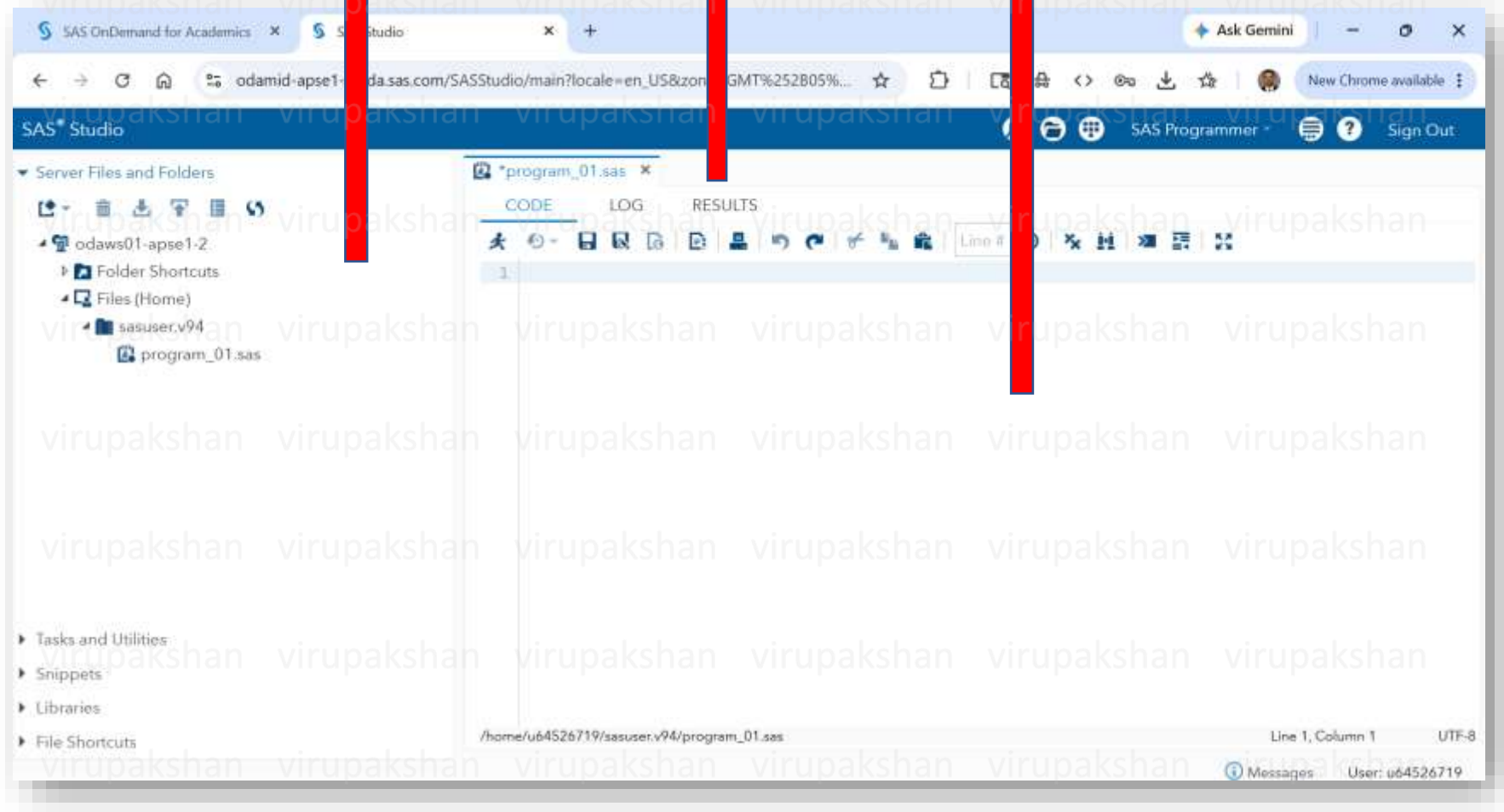
L02.3 After Login to SAS Studio , click **Launch**

<https://welcome.oda.sas.com/>



L02.4 After Launch,

- Left panel → Files/Folders
- Center area → Code Editor
- Bottom → Log and Results



LAB TASK 03: Write Your First SAS Program



L03.1 Write your first **code** (text)

```
/* My First SAS Program */
```

```
data students;
```

```
  input Name $ Age Marks;
```

```
  datalines;
```

```
Arun 20 85
```

```
Priya 21 92
```

```
Kumar 19 78
```

```
Divya 22 88
```

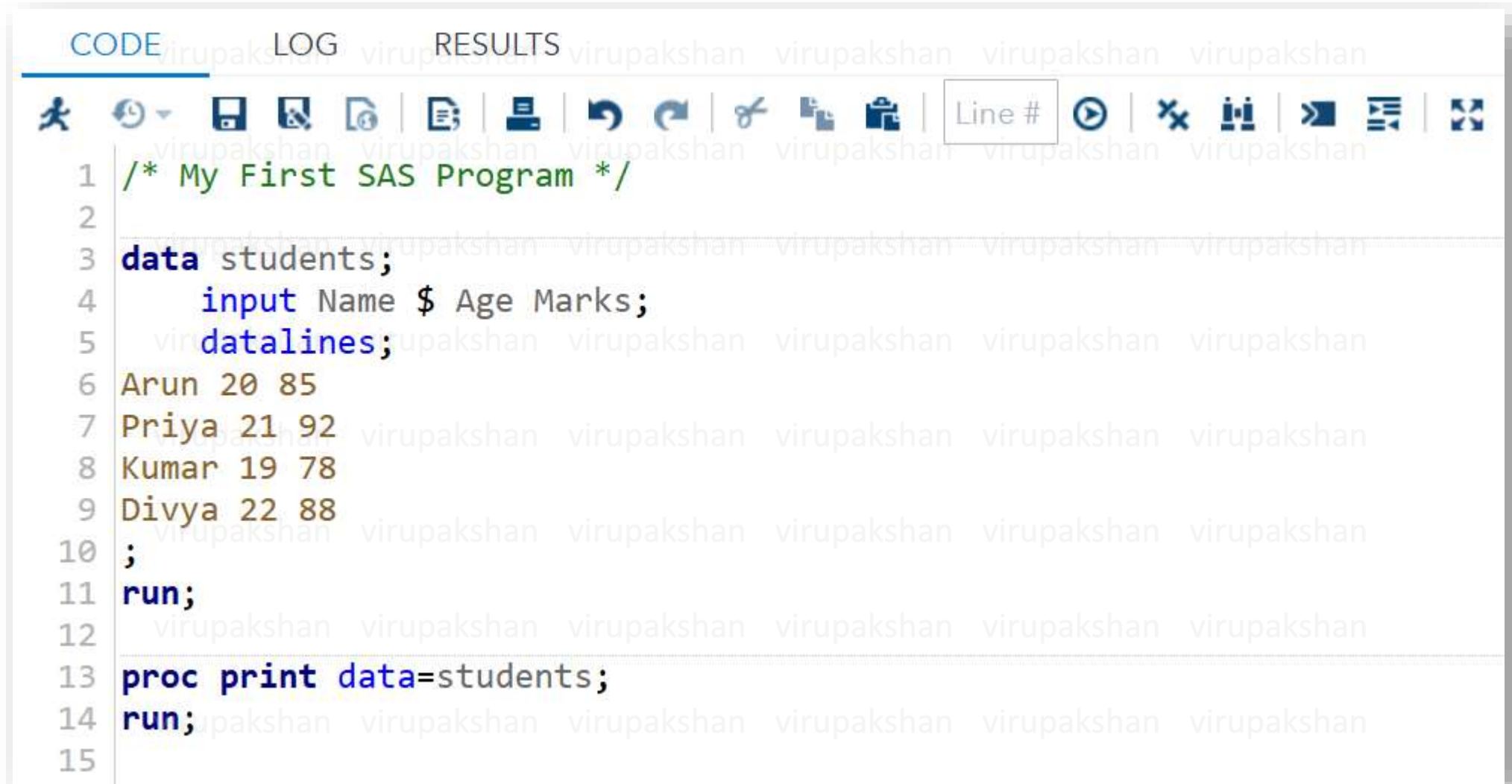
```
;
```

```
run;
```

```
proc print data=students;
```

```
run;
```

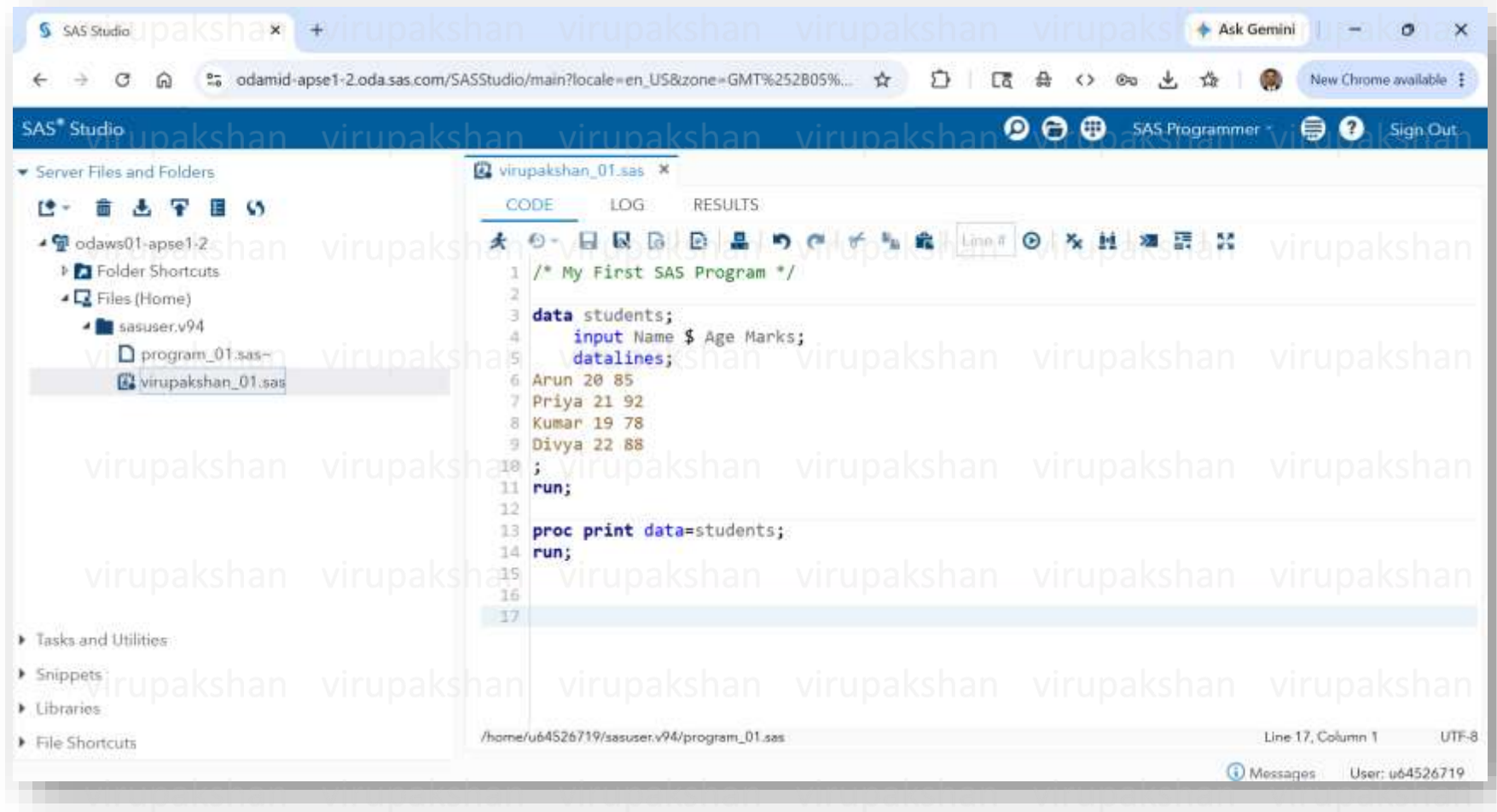
L03.1b Write your first code



The screenshot shows a SAS IDE window with three tabs: CODE, LOG, and RESULTS. The CODE tab is active, displaying a SAS program. The program starts with a comment line, followed by a DATA statement to create a dataset named 'students'. The INPUT statement defines the variables: Name (character), Age (numeric), and Marks (numeric). The DATALINES statement indicates the start of the data. The data consists of four rows: Arun (Age 20, Marks 85), Priya (Age 21, Marks 92), Kumar (Age 19, Marks 78), and Divya (Age 22, Marks 88). The program ends with a PROC PRINT statement to display the dataset and a RUN statement to execute the program. The IDE interface includes a toolbar with various icons for file operations, editing, and execution. A 'Line #' box is visible in the toolbar.

```
1 /* My First SAS Program */
2
3 data students;
4     input Name $ Age Marks;
5     datalines;
6 Arun 20 85
7 Priya 21 92
8 Kumar 19 78
9 Divya 22 88
10 ;
11 run;
12
13 proc print data=students;
14 run;
15
```

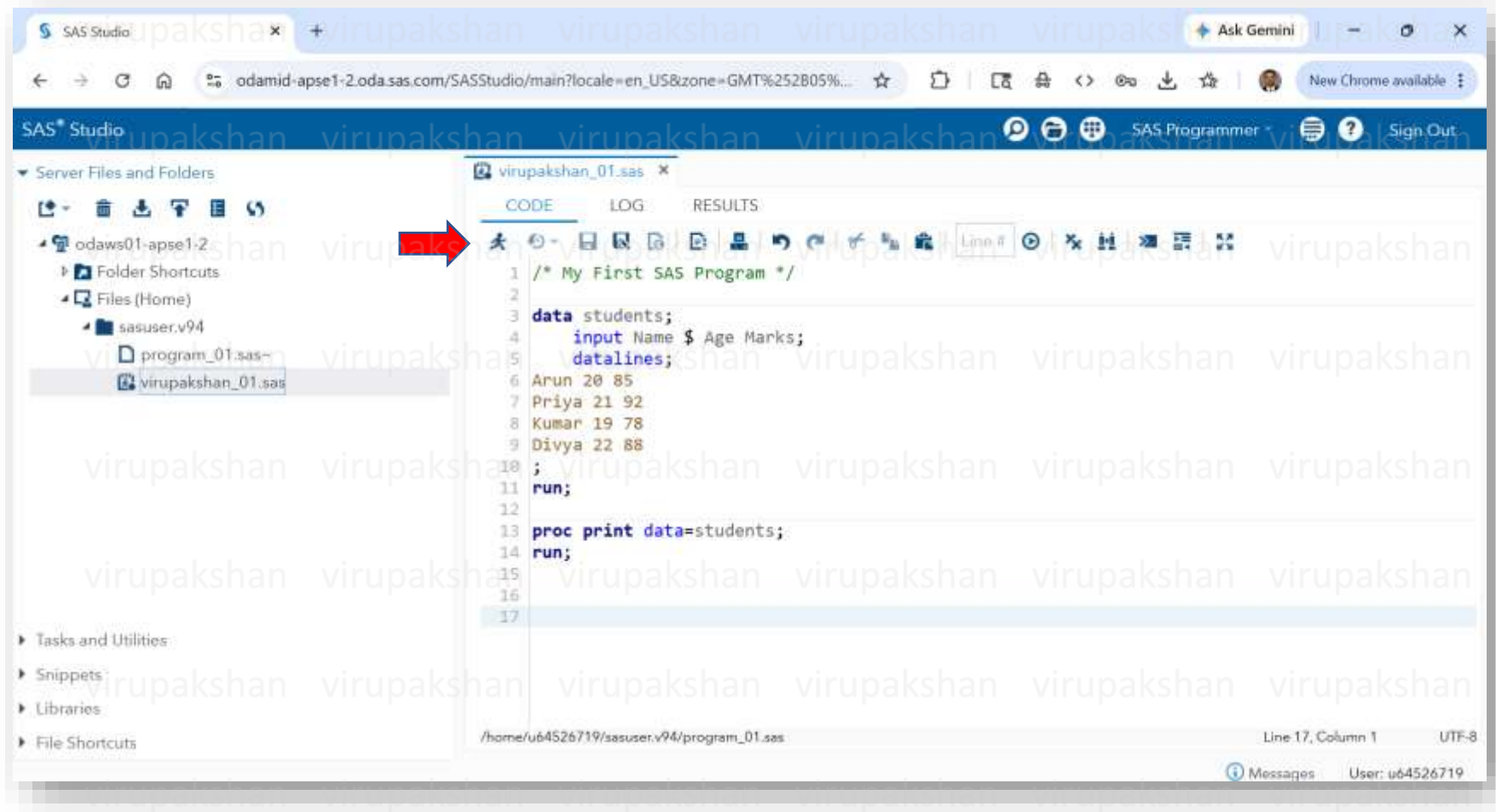
L03.2 Save as “yourName_01.sas”



L03.3 Press **RUN**



in “yourName_01.sas”



L03.4 See **RESULTS** in “yourName_01.sas”

The screenshot displays the SAS Studio web interface. The left sidebar shows the 'Server Files and Folders' pane with a tree structure: 'odaws01-apse1-2' > 'Folder Shortcuts' > 'Files (Home)' > 'sasuser.v94' > 'program_01.sas~' > 'virupakshan_01.sas'. The main workspace has tabs for 'CODE', 'LOG', 'RESULTS', and 'OUTPUT DATA'. The 'RESULTS' tab is active, showing a 'Table of Contents' and a data table. The data table has columns 'Obs', 'Name', 'Age', and 'Marks' with 4 rows of data. The bottom status bar indicates 'Messages: 1' and 'User: u64526719'.

Obs	Name	Age	Marks
1	Arun	20	85
2	Priya	21	92
3	Kumar	19	78
4	Divya	22	88

L03.5 See **OUTPUT DATA** in “yourName_01.sas”

The screenshot displays the SAS Studio web interface. The left sidebar shows the 'Server Files and Folders' pane with a tree view containing 'odaws01-apse1-2', 'Folder Shortcuts', 'Files (Home)', 'sasuser.v94', 'program_01.sas~', and 'virupakshan_01.sas'. The main workspace has tabs for 'CODE', 'LOG', 'RESULTS', and 'OUTPUT DATA', with 'OUTPUT DATA' being the active tab. Below the tabs, a dropdown menu shows 'Table: WORK.STUDENTS' and 'View: Column names'. A 'Columns' panel on the left lists 'Select all', 'Name', 'Age', and 'Marks', all of which are checked. Below this is a 'Property Value' table with rows for 'Label', 'Name', 'Length', and 'Type'. The main area displays a data table with 4 rows and 3 columns: 'Name', 'Age', and 'Marks'. The data is as follows:

	Name	Age	Marks
1	Arun	20	85
2	Priya	21	92
3	Kumar	19	78
4	Divya	22	88

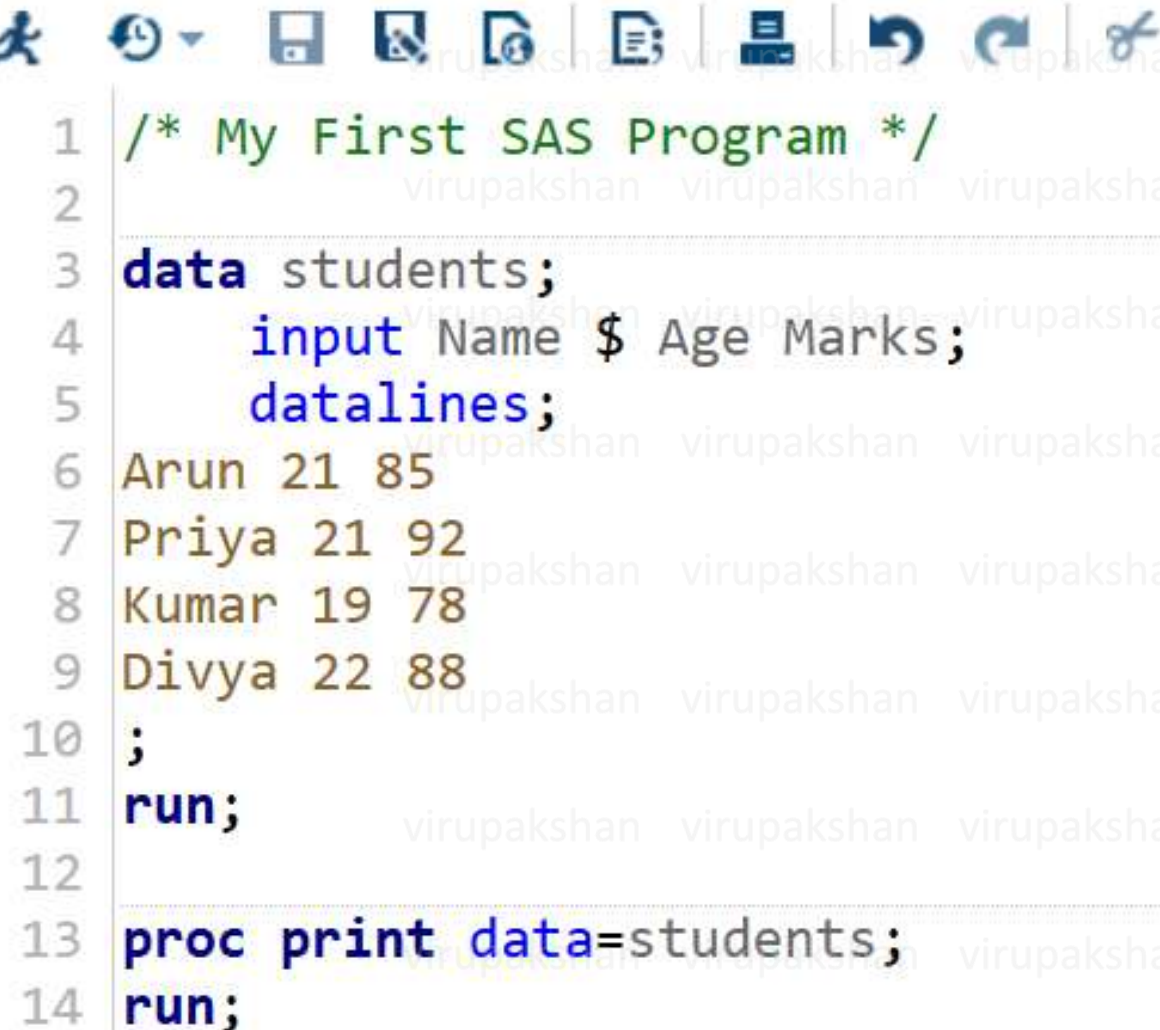
At the bottom of the interface, a status bar shows 'Messages: 1' and 'User: u64526719'.

LAB TASK 04: Understand Your First SAS Program



L04.1.0 Understand **code**

VirupakshanK@gmail.com



```
1 /* My First SAS Program */
2
3 data students;
4     input Name $ Age Marks;
5     datalines;
6 Arun 21 85
7 Priya 21 92
8 Kumar 19 78
9 Divya 22 88
10 ;
11 run;
12
13 proc print data=students;
14 run;
```

VirupakshanK@gmail.com

L04.1.0 Understand your **code**

```
/* My First SAS Program */
```

```
data students;
```

```
  input Name $ Age Marks;
```

```
  datalines;
```

```
Arun 20 85
```

```
Priya 21 92
```

```
Kumar 19 78
```

```
Divya 22 88
```

```
;
```

```
run;
```

```
proc print data=students;
```

```
run;
```

L04.1.1 Understand **comment**

VirupakshanK@gmail.com

```
1  /* My First SAS Program */
2
3  data students;
4      input Name $ Age Marks;
5      datalines;
6  Arun 21 85
7  Priya 21 92
8  Kumar 19 78
9  Divya 22 88
10 ;
11 run;
12
13 proc print data=students;
14 run;
```

COMMENT

Single line:

```
</> sas
```

```
* This is a comment;
```

Block comment:

```
</> sas
```

```
/* This is a comment */
```

L04.1.2 Understand data

VirupakshanK@gmail.com

data

Creates a **dataset** called **students**

```
1  /* My First SAS Program */
2
3  data students;
4      input Name $ Age Marks;
5      datalines;
6  Arun 21 85
7  Priya 21 92
8  Kumar 19 78
9  Divya 22 88
10 ;
11 run;
12
13 proc print data=students;
14 run;
```

L04.1.3 Understand variables

VirupakshanK@gmail.com

```
1  /* My First SAS Program */
2
3  data students;
4      input Name $ Age Marks;
5      datalines;
6  Arun 21 85
7  Priya 21 92
8  Kumar 19 78
9  Divya 22 88
10 ;
11 run;
12
13 proc print data=students;
14 run;
```

input Name \$ Age Marks;

Defines variables:

Variable	Type
Name	Character (\$)
Age	Numeric
Marks	Numeric

L04.1.4 Understand datalines

```
1  /* My First SAS Program */
2
3  data students;
4      input Name $ Age Marks;
5      datalines;
6  Arun 21 85
7  Priya 21 92
8  Kumar 19 78
9  Divya 22 88
10 ;
11 run;
12
13 proc print data=students;
14 run;
```

datalines

Used to enter data manually.

L04.1.5 Understand run

VirupakshanK@gmail.com

```
1  /* My First SAS Program */
2
3  data students;
4      input Name $ Age Marks;
5      datalines;
6  Arun 21 85
7  Priya 21 92
8  Kumar 19 78
9  Divya 22 88
10 ;
11 run;
12
13 proc print data=students;
14 run;
```

run;

Ends the step.

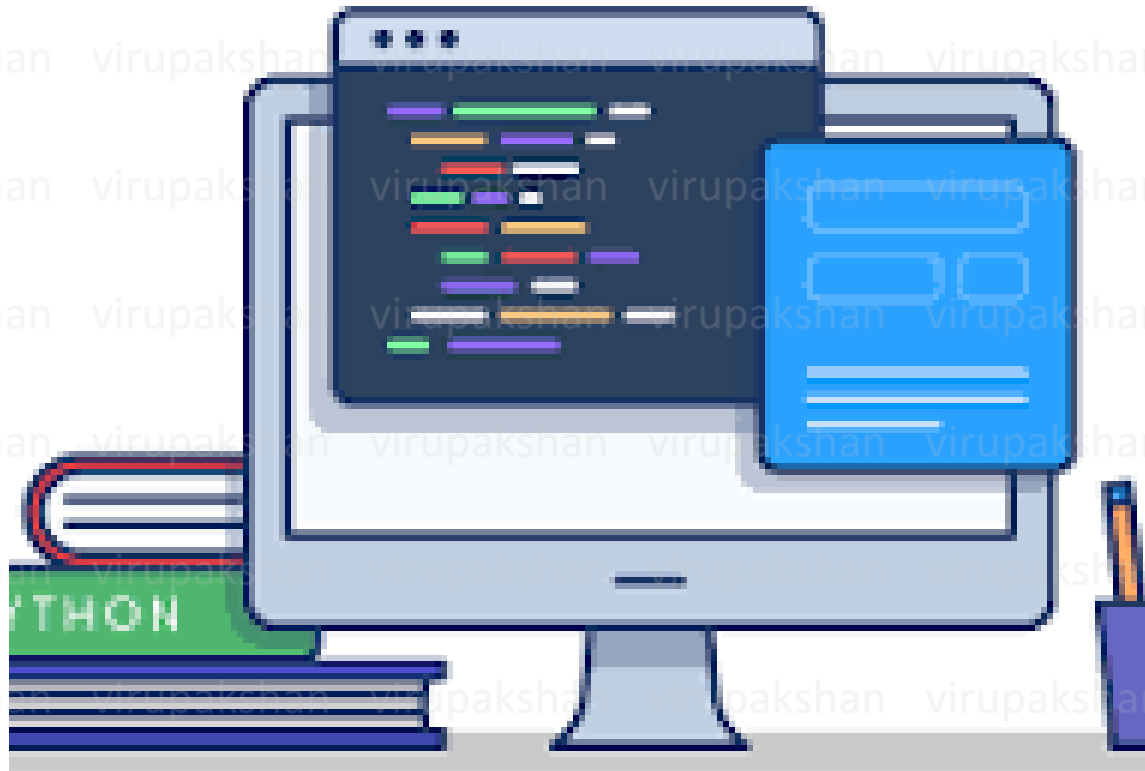
L04.1.6 Understand PROC PRINT

```
1  /* My First SAS Program */
2
3  data students;
4      input Name $ Age Marks;
5      datalines;
6  Arun 21 85
7  Priya 21 92
8  Kumar 19 78
9  Divya 22 88
10 ;
11 run;
12
13 proc print data=students;
14 run;
```

proc print data=students;
run;

Displays the dataset as a table.

LAB TASK 05: Understand basics of SAS Program

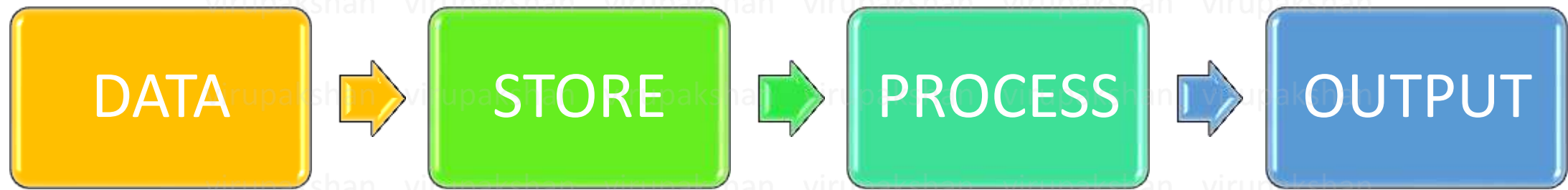


5.1 Understand SAS program : DATA and PROC

Every SAS program usually has:

Section	Purpose
DATA Step	Create/modify data
PROC Step	Analyze/display data

5.2 Understand SAS process



5.3 Understand SAS syntax basics

1. Statements end with semicolon

Correct:

```
</> sas
```

```
x = 10;
```



Wrong:

```
</> sas
```

```
x = 10
```



2. Comments

Single line:

```
</> sas
```

```
* This is a comment;
```

Block comment:

```
</> sas
```

```
/* This is a comment */
```

3. RUN statement

Executes the step.

```
</> sas
```

```
run;
```

5.4 What is the difference between RESULTS & OUTPUT DATA?

Feature	RESULTS	OUTPUT DATA
Purpose	Report	Dataset
Generated by	PROC	DATA step
Format	Formatted output	Raw data
Editable	No	Sometimes
Similar to	PDF report	Excel table

5.4.2 What is the difference between RESULTS & OUTPUT DATA?

The **RESULTS** tab shows the **formatted report/output** produced by PROC steps.

Example:

```
</> sas  
  
proc print data=dset;  
run;
```

This creates a report/table view in the RESULTS tab.

5.4.3 What is the difference between RESULTS & OUTPUT DATA?

The **OUTPUT DATA** tab shows the **actual dataset stored in memory/work library**. It is more like viewing raw data.

OUTPUT DATA Characteristics

Feature	Description
Purpose	Shows dataset contents
Looks like	Spreadsheet/data grid
Generated by	DATA step
Editable?	Sometimes yes
Similar to	Excel sheet/database table

5.4.4 What is the difference between RESULTS & OUTPUT DATA?

RESULTS

Shows:

“How SAS prints the data”

Obs	Name	Score
1	Arun	90
2	Priya	95

OUTPUT DATA

Shows:

“What SAS actually stored”

The screenshot shows the SAS Results window for the file 'Results_vs_OutputData_03.sas'. The 'OUTPUT DATA' tab is selected. The table is 'WORK.MARKS'. The columns are 'Name' and 'Score'. The data is displayed in a table with two rows: 1 Arun 90 and 2 Priya 95. The 'Columns' list shows 'Select all', 'Name', and 'Score' are selected. The 'Property' and 'Value' table shows the variable properties: Label Name, Name Name, Length 8, Type Char, Format, and Informat.

Columns	Property	Value
☒ Select all	Label	Name
☒ Name	Name	Name
☒ Score	Length	8
	Type	Char
	Format	
	Informat	

	Name	Score
1	Arun	90
2	Priya	95

5.4.5 What is the difference between RESULTS & OUTPUT DATA?

```
1 data marks;
2   input Name $ Score;
3   datalines;
4 Arun 90
5 Priya 95
6 ;
7 run;
8
9 proc print data=marks;
10 run;
```

DATA Step

Creates dataset:

WORK.MARKS

This appears in:

- OUTPUT DATA

PROC PRINT

Creates formatted display.

This appears in:

- RESULTS

OUTPUT DATA

Shows:

"What SAS actually stored"

T... Rows 1-2	
Name	Score
1 Arun	90
2 Priya	95

RESULTS

Shows:

"How SAS prints the data"

Obs	Name	Score
1	Arun	90
2	Priya	95

5.4.6 What happens when you comment **proc print**?

```
1 data marks;  
2   input Name $ Score;  
3   datalines;  
4 Arun 90  
5 Priya 95  
6 ;  
7 run;  
8  
9 /* proc print data=marks;  
10 run; */
```



DATA Step

Creates dataset:

WORK.MARKS

This appears in:

- OUTPUT DATA

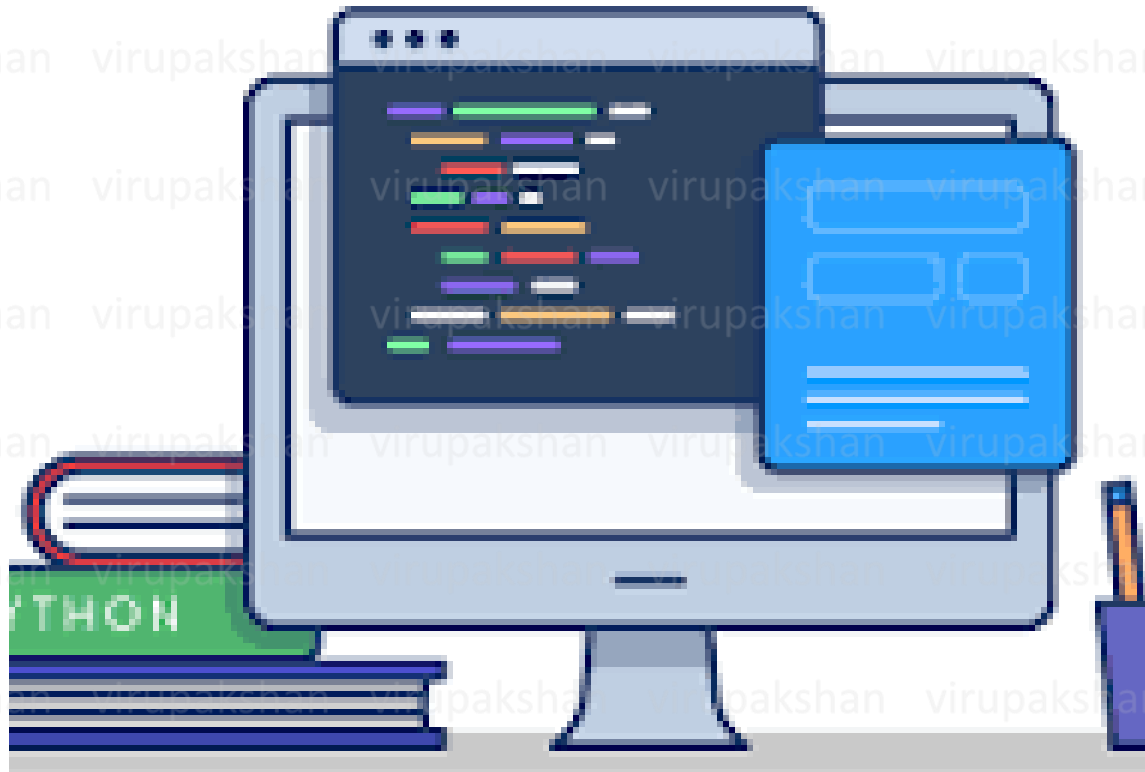
PROC PRINT

Creates formatted display.

This appears in:

- RESULTS

LAB TASK 06: SAS Program – Simple Arithmetic



L06.1 Simple Arithmetic Program

```
1 /* A simple arithmetic program */
2 data dset;
3   input Name $ Tamil English Maths;
4   Total = Tamil + English + Maths;
5   Average = Total / 3;
6   datalines;
7 Arun 80 75 100
8 Priya 95 88 92
9 Kumar 94 89 92
10 ;
11 run;
12
13 proc print data=dset;
14 run;
```

viru_arithmetic_02.sas

CODE LOG RESULTS OUTPUT DATA

Table of Contents

Obs	Name	Tamil	English	Maths	Total	Average
1	Arun	80	75	100	255	85.0000
2	Priya	95	88	92	275	91.6667
3	Kumar	94	89	92	275	91.6667

LAB TASK 07: SAS Program – Simple Arithmetic – Play with it!



L07.1 What happens if **run;** is changed to *** run;** ?

```
1  /* A simple arithmetic program */
2  data dset;
3      input Name $ Tamil English Maths;
4      Total=Tamil + English + Maths;
5      Average=Total / 3;
6      datalines;
7  Arun 80 75 100
8  Priya 95 88 92
9  Kumar 94 89 92
10 ;
11 * run;
12
13 proc print data=dset;
14 * run;
```



L07.2 If **run;** is changed to *** run;** the program still runs.

```
1 /* A simple arithmetic program */
2 data dset;
3   input Name $ Tamil English Maths;
4   Total=Tamil + English + Maths;
5   Average=Total / 3;
6   datalines;
7 Arun 80 75 100
8 Priya 95 88 92
9 Kumar 94 89 92
10 ;
11 * run;
12
13 proc print data=dset;
14 * run;
```

CODE

LOG

RESULTS

OUTPUT DATA



► Table of Contents

Obs	Name	Tamil	English	Maths	Total	Average
1	Arun	80	75	100	255	85.0000
2	Priya	95	88	92	275	91.6667
3	Kumar	94	89	92	275	91.6667

Why?

L07.3 If **run;** is changed to *** run;** the program still runs.

Because...

Important SAS Concept

In SAS, many steps end automatically when:

- another **DATA** step starts
- another **PROC** step starts
- program ends

This is called an **implicit RUN**.

Then Why Do We Write **RUN;** ?

Because it is:

- clearer
- safer
- professional coding practice
- easier for debugging
- necessary in many advanced programs

LAB 01

21CSE295T

Statistical Data Analytics And Integration

Doubts?

THANK YOU!

Virupakshan K

+91-9840847872 | VIRU@KART.LA