

# CC PROGRAMS

## PROGRAM:1

### GOOGLE FORM

#### Procedure:

1. On your web browser , login to your gmail id.
2. On the top right corner , click on the google apps icon(::) and select forms app.
3. To create a new form , either click on 'blank' or browse through the templates under 'start' a new form.
4. Rename the form by clicking on the top left corner space named as 'untitled form'.
5. The title and description of the form can be provided under the 'questions tab'.
6. To add the survey questions , under the 'questions' tab , do the following:
  - Click 'untitled question' and type the question to be asked , you can add questions by clicking on + (add question) icon.
  - To change the questions type , click on the down arrow option on the right side and you can choose one among the following types:  
  
[short answer ,paragraph , multiple choice , checkboxes, dropdown, file upload, linear scale, multiple choice grid, checkbox grid, date, time.]
  - Add response options (for example: option1=A, option 2=B, and so on) based on selected question type.
  - To delete a question, click on delete button.
  - Click on 'required' option if the response is mandatory.
7. To edit questions do the following:
  - Drag to reorder the questions.
  - Click more to add a description and response validation .
8. Response validation puts a conditions on the type of response to be written.
  - To add an image to question , click on 'add image' icon.
  - To add a video to question , click on 'add video' icon.
  - To add a section, click on 'add section' icon on the right side.
9. To send the form to other users, click on the 'send button' in the top right corners , you can send your response via email , link ,embed html options .
10. To know the response click on the 'response' tab in the form. The responses can be viewed on different formats.
  - 'summary' option gives graphical representation of the responses to each question , it can be in the form of a bar graph , pie chart etc.
  - 'question' option gives all the responses to each question . we can select a particular question is the drop down or we can slide through each question.
  - 'individual' option is a slide through option where response to all the questions by each user can be obtained.
  - The responses can be downloaded as a 'csv' file . print all the responses and delete all responses options available.

11. The 'settings' tab provides ways to handle 'questions', 'responses', presentation with form defaults and question defaults, used for creating the form.

## GOOGLE SHEETS

### Procedure:

1. Go to google apps on your web browser and click on google sheets .
2. Click on 'blank' to get a new untitled spread sheet and rename the file.
3. To Create the 'COURSE OUTCOME CALCULATION' information through spread sheets, insert in the cells, college name, branch name, semester, subject name, subject code, and title.
4. The columns in the sheets should have serial number, USN, CIE1 marks, along with course outcome(CO1, CO2, CO3) for the particular question(1a, 1b, 2a, 2b, 2c) that it belongs to and similarly CIE2 marks, then assignment 1 marks along with course outcome and similarly assignment 2 marks.
5. The next 3 columns are CO1, CO2, CO3, which are sum of marks obtained in each course outcome through CIE and assignment marks.  
FORMULA:  $\text{SUM}(\text{E16}, \text{F16}, \text{G16}, \text{Q16})$   
Where E16, F16, G16, Q16 are particular cell numbers .
6. We can apply this formula for all the rows by clicking on the button right corner of the cell and dragging it to all rows needed. Formulas are typed in that particular cells
7. The average of each course outcome (75% of average) is calculated as:  
FORMULA:  $=0.75 * \text{AVERAGE}(\text{T16}:\text{T30})$   
Where T16:T30 are the number of rows to which the formula is applied in T column .the calculated value will appear in the cell.
8. The target level is set as LEVEL 1(<40%), LEVEL 2(>40%) and (<75%) and LEVEL 3(>75%)
9. The total students for each course outcome is (CO1, CO2, CO3) is inserted.
10. Students above target for each course outcome is calculated using :  
FORMULA:  $\text{COUNT IF}(\text{T16}:\text{T30}, ">=" \& \text{T13})$   
This is to check if each individual student has scored above the average calculate step 7 or not) . the calculated value appear in the cell.
11. To calculate co assignment(%) we use the formula, FORMULA= $= (\text{E43}/\text{E42} * 100)$   
E43->students above target  
E42-> total students  
The calculated value will appear in the cell
12. The CO assignment % is used to calculate the CO assignment level, using:  
FORMULA:  
 $=\text{IF}(\text{E44}>75, "3", \text{IF}(\text{AND}(\text{E44}>40, \text{E44}<75), "2", \text{IF}(\text{E44}<40, "1", "0")))$   
Where, E44->CO ASSESSMENT(%)
13. We can calculate the above formulas for one course outcome, suppose CO's and then click on the button right corner of the cell and drag it to the columns CO2 and CO3 for each formul.
14. All the changes made are saved automatically .

## **PROGRAM:2**

### **SALESFORCE -PRODUCT INFORMATION**

#### **PROCEDURE**

1. Go to [trailhead.salesforce.com](https://trailhead.salesforce.com) on your browser
2. Create an account on salesforce
3. Login to salesforce and click on your profile icon on the top right corner and click on hands-on-org
4. Choose your playground and click on launch
5. Click on setup icon on the top right corner and select setup
6. Once the setup window appears, then select the quick find box and search app manager and click on it

7. Click on new lighting app on the right corner to create a new application

8. When the new lighting app window appears, fill in the app details

Eg- App name: Inventory

Developer name: Rashmi

Description: the app contains product information

Then click on next and next

9. Select the utility items and navigation items required for your app & click on next

10. Choose the user profile , select system administrator that can access the app .click on save and finish .Your app is created and can be view in app manager

11. Click on app launcher, search for your app and click on it, A new window appears where you can view the features, tabs in your app

12. Go to object manager (in setup) and click on new object. Create a new custom object with label as product click on save

13. Click on object manager and select the created custom object ,product , select fields and relationships option and click on new.

The fields are as follows

Field Data type

Product\_id autonumber

Product\_name text

Description text area

Quantity number

Price number

Purchase\_date date

14. Click on home, In the quick find box, search tabs and click on it

15. Click on new, in the custom object tabs and select a tab style, click on next and the next.

Under Include tab click your custom app and click save

16. Now you can view the custom object tab, product in your app, Inventory

17. Click on new, to add records under the object, 'product'.

## PROGRAM:3

### Master-Child relationship

#### Procedure:

1. Go to trailhead.salesforce.com on your web browser. Create an account and login.
2. Create an App, Inventory.
3. Go to setup. Click on object manager. Click new object.
4. Create a custom object named, Product.
5. Go to object manager, select the object, Product. Go to Fields and Relationship and add the following fields:

<u>Field</u>	<u>Data Type</u>
Product_id	auto number
Product_name	text
Description	text area
Quantity	number
Price	number
Purchase_date	date

6. Create another custom object named, Invoice.

7. Go to object manager, select the object, Invoice. Click on Fields and Relationship and add the following fields:

<u>Field</u>	<u>Data Type</u>
Invoice_id	auto number
Invoice_name	text
Address	text area
Date	date
Mail	email

8. Create another(new) field under Invoice's Fields and Relationship with data type as Master-Detail Relationship. Click next.
9. Select the object to which this object is related to, that is, in the drop down, select, product. Click on next and then save.
10. Go to Home. In the quick find box , search tabs and click on it.
11. Under custom object Tabs, click new, select object 'product' and tab style. Click next and save.
12. Under custom object tabs, click new, select object 'invoice' and tab style. Click next and save.
13. Go to App Launcher and select your app. Inventory.

14. When app window appears, add records into object, Product by clicking on new under the products tab.

15. To add records into the object 'Invoice', Click on new, under the invoice tab. Now, select the particular product record from the drop down list, to enter Invoice details for that record (Product).

#### PROGRAM:4

**Develop a visual force page to demonstrate working of basic visual components.**

#### PROCEDURE:

- 1]Go to trailhead salesforce.com on your browser and login to your salesforce account.
- 2]Click on your profile icon present in the top right corner and select Hands-on-orgs. Launch the playground.
- 3]Click on the setup icon in the top right corner and select Developer Console.
- 4]Click on the file in the Developer Console window.

Click New>>Visual Force Page

- 5] Enter the filename as index.vfp

Enter the following code :

```
<apex:page>
    <h1> intro</h1>
    <p> { ! $ User.FirstName & ' ' & $ User.LastName} </p>
    <p> The year today is { !YEAR((TODAY()))} </p>
    <p> The date tomorrow will be { ! DAY(TODAY()+1)} </p>
    <p> Lets find maximum : {!MAX(1,2,81,9,80)} </p>
    <p> The square root of 81 is : {!SQRT(81)}</p>
</apex: page>
```

- 6] Save the file and click on Test>>Run All
- 7] Repeat Steps 5 and 6 for the programs index1.vfp and index2.vfp

Index1.vfp

```
<apex:page standardController="Account">
    <apex:pageBlock title="User Details">
        <apex:pageBlockSection title="Expand">
            {!$User.FirstName } {!$User.LastName }
            ({!$User.Username})
        </apex:pageBlockSection>
    </apex:pageBlock>
    <apex:pageBlock title="Morning Session">
```

```

    <apex:pageBlockSection title="Expressions" columns="1">
    {! $User.FirstName & ' ' & $User.LastName }
    <p>The year today is {!YEAR(TODAY())}</p>
    <p>Tomorrow will be day number {!DAY(TODAY() + 1 )}</p>
    <p>Let's find a maximum: {!MAX(1,2,3,4,5,6,85,4,3,2,1)}</p>
    <p>The square root of 81 is {!SQRT(81)}</p>
    <p>Is it true? {!CONTAINS('salesforce.com','force.com')}</p>
    <p> {!IF(CONTAINS('salesforce.com','force.com'),
        'Yes','No')}</p>
    <p> {!IF(DAY(TODAY()) < 4,
        'Before the 4th' , 'The 4th or after')}</p>
    </apex:pageBlockSection>
</apex:pageBlock>
<apex:pageBlock title="Afternoon Session">
    <apex:pageBlockSection title="Account Summary">
        Name: {!Account.Name } <br/>
        Phone: {!Account.Phone } <br/>
        Industry: {!Account.Industry } <br/>
        Revenue: {!Account.AnnualRevenue } <br/>
    </apex:pageBlockSection>
</apex:pageBlock>
<apex:pageBlock title="Contacts">
    <apex:pageBlockTable value="{! Account.contacts}" var="contact">
        <apex:column value="{!contact.Name}"/>
        <apex:column value="{!contact.Title}"/>
        <apex:column value="{!contact.Phone}"/>
    </apex:pageBlockTable>
</apex:pageBlock>
</apex:page>

```



```
<apex:page standardController="Account">
  <apex:form >
    <apex:pageBlock title="Edit Account">
      <apex:pageBlockSection >
        <apex:inputField value="{! Account.Name}"/>
        <apex:inputField value="{! Account.Phone}"/>
        <apex:inputField value="{! Account.Industry}"/>
        <apex:inputField value="{! Account.AnualRevenue}"/>
      </apex:pageBlockSection>
      <apex:pageBlockButtons >
        <apex:commandButton action="{! save}" value="Savelabel"/>
      </apex:pageBlockButtons>
    </apex:pageBlock>
  </apex:form>
</apex:page>
```

8] Click on the Preview to the output of the program.

9] For all the program, index1.vfp , enter the account details, Click on the Save Label.

10] Now, copy the id of that particular record from the 'url' above.

11] Click on the Preview to the output of index2.vfp.

12] In the url, paste the id copied previously and press enter to see the account details on the page.

## **PROGRAM:5**

### **PROCEDURE:**

1. Create a account in code anywhere by providing your email-id and password through sign up option.
2. Verify your account by clicking on the link ,sent to the given email id.
3. Create a container in the dashboard by clicking on “new container”.
4. Select python and click on create.
5. Once the container is created , as IDE workspace will open.
6. Create new project by clicking on new folder option and name it as :project”.
7. Create a new file named ‘dbprogram.py’ by clicking on new file option.
8. Open ‘new tab’ and go to official website of Codeanywhere. Then click on ‘Dashboard’ and go to ‘Advanced Topics’ ,Now click on “Installing MySQL”.
9. To install MySQL in your container, you have to run the following commands in the IDE terminal:

```
Sudo apt-get update
```

```
Sudo apt-get install mysql-server
```

```
Sudo service MySQL status
```

10. Once the installation is done,type the following command

```
Mysql—user=root—password=root
```

11. Now create a new database named ‘COURSEMATCH’ using the following command:

```
Create database COURSEMATCH;
```

```
Use COURSEMATCH;
```

12. Now create a new table in database using the following Command:

```
Create table course_details(c_id varchar(20), stream varchar(20),c_name varchar(20));
```

13. Insert values into the table using the following command:

```
Insert into course_details values('01','ISE','CC')
```

14. To display the table, use the following commands:

```
Select * from course_details;
```

15. Prerequisites to run the program:

Download mysql-connector module

```
pip install mysql-connector-python
```

Download pandas module

```
pip install pandas
```

16. Now, in the file 'dbprogram.py', write the following code:

```
import os
import platform
import mysql.connector
import pandas as pd
mydb=mysql.connector.connect(host="localhost",\
                             user="root",\
                             passwd="root",\
                             database="COURSE_MATCH")
mycursor=mydb.cursor()

def course_Insert():
    L=[]
    c_id=int(input("Enter the course id : "))
    L.append(c_id)
    stream=input("Enter The Stream Name:")
    L.append(stream)
    c_name=input("Enter available opportunities or courses in this stream : ")
    L.append(c_name)
    course=(L)
    sql="insert into course_details (c_id, stream, c_name) values (%s,%s,%s)"
    mycursor.execute(sql,course)
    mydb.commit()

def cView():
    print("Select the search criteria : ")
    print("1. c_id")
    print("2. Stream")
    print("3. All")
    ch=int(input("Enter the choice : "))
    if ch==1:
        s=int(input("c_id : "))
        c=(s,)
        sql="select * from course_details where c_id=%s"
        mycursor.execute(sql,c)
    elif ch==2:
        s=input("Enter stream Name : ")
        n=(s,)
```

```

        sql="select * from course_details where stream=%s"
        mycursor.execute(sql,n)
    elif ch==3:
        sql="select * from course_details"
        mycursor.execute(sql)
    res=mycursor.fetchall()
    print("The course details are as follows : ")
    print("course_id, Stream_Name,Course_opportunities")
    for x in res:
        print(x)

def removecourse():
    c_id=int(input("Enter the course_id of the course to be deleted : "))
    ci=(c_id,)
    sql="Delete from course_details where c_id=%s"
    mycursor.execute(sql,ci)
    sql="Delete from course_details where c_id=%s"
    mycursor.execute(sql,ci)
    mydb.commit()
def MenuSet(): #Function For The course match
    print("Enter 1 : To Add course")
    print("Enter 2 : To View course ")
    print("Enter 3 : To Remove course")
    try: #Using Exceptions For Validation
        userInput = int(input("Please Select An Above Option: ")) #Will Take Input From User
    except ValueError:
        exit("\nHy! That's Not A Number") #Error Message
    else:
        print("\n") #Print New Line
        if(userInput == 1):
            course_Insert()
        elif (userInput==2):
            cView()
        elif (userInput==3):
            removecourse()
        else: print("Enter correct choice. . . ")

MenuSet()
def runAgain():
    runAgn = input("\nwant To Run Again Y/n: ")
    while(runAgn.lower() == 'y'):
        if(platform.system() == "Windows"):
            print(os.system('cls'))
        else:
            print(os.system('clear'))
        MenuSet()
    runAgn = input("\nwant To Run Again Y/n: ")
runAgain()

```

17. In the console/terminal to run the code

Type:python dbprogram.py

18. The output will be displayed.

## PROGRAM:6

### Demonstrate the working of Git and GitHub

#### PROCEDURE:

1. Create an account on GitHub by providing your email-id and password. Give a username.
2. Verify your account by entering the code sent to your email-id.
3. To create a repository on GitHub, click on create repository option. Enter the repository name and click on create repository.
4. Open new tab and type 'git-scm.com', to download Git for Windows.
5. Create a folder on your PC with few files and directories.
6. Right click on the folder created and click on 'Git Bash Here' option.
7. Type the command:  
`git init`  
to create an empty Git repository in specified directory
8. To clone the repository created on GitHub onto the local machine, first copy the url from GitHub and then type the following command on Git console:  
`git clone <repo>`
9. Type the command:  
`git config user.name <name>`  
`git.config user.email <email>`  
to define author name and email to be used for all commits in the current repository.
10. To add a file or directory to your repository before next commit, type the command:  
`git add <file> / <directory>`
11. To list which files are staged, upstaged, and untracked, type the command:  
`git status -s`
12. To commit the files or directory, to save changes permanently, type the command:  
`git commit -m <message>`
13. To get the commit history, type,  
`git log`
14. To push the files stored in the folder on the local machine to remote GitHub repository, type,  
`git push <repo><branch>`  
repo -> url (via HTTP or SSH)
  - It creates a named branch in the remote repository if it doesn't exist.
  - Once, you click enter, we get a dialog box asking to sign in with a code, click on it.
  - A url with code will be displayed.
  - Copy the code and click on the url.
  - The url takes you to a device activation page and paste the code there. Then click on 'Authorize GitCredentialManager'.
  - Now the file is pushed onto GitHub remote repository and can be accessed anywhere with the internet and login credentials.

15. To create a git ignore files, type the command:  
    `touch .gitignore`
16. Type the filename along with extension inside the gitignore file created, and these files will be ignored when pushed to the repository.
17. To check the branch in the console,  
    Type:  
        `git checkout -b B`  
    Now the branch is B
18. Type:  
        `git push origin B`  
    A new branch named B is created in the GitHub repository.

## **PROGRAM:7**

### **Steps:**

1. Go to Oracle virtual box on your web browser. Click 'downloads' option.
2. Under Virtualbox 6.1.34 platform packages.  
Click on "Windows hosts", VM is downloaded.
3. A dialog window appears, click on Next and then 'finish'.
4. After installation, click on New on Oracle VM  
and name your file.
5. Set the memory size as required.
6. Set hard disk by clicking on 'Create disk file' now.
7. Click on disk symbol at the right corner of the window and click on 'choose disk file'. Select folder containing OS to be virtually used.
8. Double click on the machine you created in step 4 to run the machine.
9. To create a directory, use the command,  
  
    # mkdir cc
10. To list all files, use the command ,  
  
    # ls.
11. To change directory, use the command  
  
    # cd cc
12. To create a new blank file,  
  
    # touch 'File name'
13. To remove the file,  
  
    # rm 'filename'
14. #wc 'filename'  
  
This command tells how many lines, words, and characters are there in a file.
15. The grep command is used to look for strings in the file  
  
    # grep -l 'this' 'filename'
- Output: This is my first file.
16. The command to know the present working directory :  
  
    # pwd
17. To view file contents,



#cat 'filename'

18.To view the online reference manuals,

#man

19.Command to move/rename the file1 to file2:

#mv file1 file2.

20.The command to display number of bytes occupied by the contents of the directory.

#du

21.The command to display the calendar,

#cal

22.The command to display date and time,

#date

23.The command to remove a directory,

#rm cc

## PROGRAM:8

**Demonstrate file handling program on codeanywhere.**

### Procedure :

1. Create an account in codeanywhere by providing your email id and password through signup option.
2. Verify your account by clicking on the link sent to given email-id.
3. Create a new container in the dashboard by clicking on 'New container'.
4. Select python and click on create
5. Once the container is created, an IDE workspace will open.
6. Create new project by clicking on new folder option and name it 'project'.
7. Create a new file named '1.py' by clicking on new file option.
8. Type the following code in the file :

```
Def program5()
    with open("/home/cabox/workspace/merge.txt","r") as f1 :
        data=f1.readlines()
    cnt_lines=0
    cnt_A=0
    cnt_B=0
    cnt_C=0
    for lines in data :
        cnt_lines+=1
        if lines[0]=='A' :
            cnt_A+=1
        if lines[0]=='B' :
            cnt_B+=1
        if lines[0]=='C' :
            cnt_C+=1
    print("Total number of lines are:",cnt_lines)
    print("Total number of lines starting with A are:",cnt_A)
    print("Total number of lines starting with B are:",cnt_B)
    print("Total number of lines starting with C are:",cnt_C)
program5()
```

9. Save the file '1.py'.
10. Create a file named 'merge.txt' using new file option and enter text.
11. Save the file merge.txt. Copy the path of the file and paste it in the code's open() function.
12. In the console to run the code, type:

Def program6():

```
cnt=0

word_search = input('Enter the word to search:')

with open("/home/cabox/workspace/merge.txt","r") as f1:
    for data in f1:
```

```
words=data.split()

for word in words:

    if(word==word_search)

        cnt+=1

print(word_search,"found",cnt,"times from the file")

return cnt

cnt1=program6()

with open("/home/cabox/workspace/merge.txt","r") as f2:

    f2.write(str(cnt1))
```

16. Save the file '2.py'
17. Create a file named 'merge.txt' using new file option and enter the text.
18. Save the file 'merge.txt' Copy the path of the file and paste it in the open() function.
19. In the console to run the code:  
Type, python 2.py
20. The output will be displayed in the console and a new file result.txt is created inside which the output will be written.

## PROGRAM:9

**Demonstrate a procedure to launch virtual machine using TryStack to execute the Bash program.**

### PROCEDURE:

1. Browse to the official Git website: <https://git-scm.com/downloads>
2. Click the download link for Windows and allow the download to complete.
3. Browse to the download location (or use the download shortcut in your browser). Double-click the file to extract and launch the installer.
4. Allow the app to make changes to your device by clicking **Yes** on the User Account Control dialog that opens.
5. Review the GNU General Public License, and when you're ready to install, click **Next**.  
The installer will ask you for an installation location. Leave the default, unless you have reason to change it, and click **Next**.
6. The next step allows you to choose a different name for your initial branch. The default is 'master.' Unless you're working in a team that requires a different name, leave the default option and click **Next**.
7. This installation step allows you to change the **PATH environment**. The **PATH** is the default set of directories included when you run a command from the command line. Leave this on the middle (recommended) selection and click **Next**.
8. Now open Git BASH , type **VI <filename.sh>** and hit enter
9. Enter **i** to enter the insert mode in VI editor
10. The navigation options are **h ,i, j , k** to navigate **left, right, down** and **up** respectively
11. **Esc** key is used to come out of the current mode
12. **:w , :q , :wq** , are the commands to write, quit, and write and quit respectively
13. Enter VI editor, enter **i** to insert the program, hit **esc** key and enter **:wq** to write and quit the editor
14. On the Bash User Interface , type the following command to run the script  
**bash filename.sh**
15. The file runs and gives the output of the program.

### Program :

```
#!/usr/bin/bash
Year=`date +%Y`
Month=`date +%m`
Day=`date +%d`
Hour=`date +%H`
Minute=`date +%M`
Second=`date +%S`
echo `date` >> book.txt
echo "Current Date is: $Day-$Month-$Year" >> book.txt
echo "Current Time is: $Hour:$Minute:$Second" >> book.txt
file='book.txt'
while read line; do
    echo $line
done < $file
```

**Result :** The expected output is obtained from the program, i.e, the date , time , hours and other specifications is met.

**Output :**

Sat Jul 30 10:41:54 IST 2022

Current Date is: 30-07-2022

Current Time is: 10:41:54

## PROGRAM:10

### PROCEDURE:

1. Install nodejs from nodejs.org
2. Go to docs.docker.com
3. Click on "Download and Install"
4. Select "Docker Desktop for Windows"

// Implementation

5. Check if docker is installed by running the following command in command prompt

```
`docker version`
```

6. Create a new folder locally
7. Create app.js file in the created folder
8. In the app.js file, insert the following code

```
`console.log("Hello World!")`
```

9. Open command prompt in the current folder and enter

```
`node app.js`
```

10. Check if the output is "Hello World!"
11. Now, create a new file in the same directory called "Dockerfile"

Note that there is no extension for this file.

12. In the Dockerfile, enter the following commands:

```
FROM node:alpine
```

```
COPY . /app
```

```
WORKDIR /app
```

```
CMD node app.js
```

13. Save the Dockerfile after entering the above commands
14. The Dockerfile will be saved in .txt format. To rename the file, do the following
  1. Open command prompt in the current directory
  2. type `ren Dockerfile.txt Dockerfile` (we are renaming the filename to just Dockerfile)

15. Now, we have to build a docker image.

In the current directory, open command prompt and enter the following command

```
`docker build -t hello-docker .`
```

The "." represents that the Dockerfile is in the current directory.

16. After the build is complete enter the following command to check whether the image is created

```
`docker image ls`
```

You should be the Repository name as 'hello-docker'

17. Now, we use the following command to run the app.js file from anywhere from our computer.

```
`docker run hello-docker`
```

Once you run the above command, you should be able to see the following output:

```
"Hello World!"
```

Alternative way to execute online

1. Go to "<https://labs.play-with-docker.com>"

2. Login with github

3. Click on start

4. Click on "Add new Instance"

5. In the terminal of the instance, type the following commands

```
`docker pull codewithmosh/hello-docker`
```

6. Enter the following command to check if the docker image was successfully pulled

```
`docker image ls`
```

You will be able to see the repository name as "codewithmosh/hello-docker"

7. Now, run ``docker run codewithmosh/hello-docker``

8. You will be able to see the output as "Hello World!"