



SREE VIDYANIKETHAN ENGINEERING COLLEGE
(AUTONOMOUS)

Sree Sainath Nagar, Tirupati

Department of Computer Science and Systems Engineering

Supporting Document for 1.1.2

Syllabus Revision carried out in 2020

Program: B.Tech.- Computer Science and Systems Engineering

Regulations: SVEC-20

This document details the following:

1. Courses where syllabus has been changed 20% and more.
2. Course-wise revised syllabus with changes highlighted.

Note: For SVEC-20 revised syllabus, SVEC-19 (previous syllabus) is the reference.

List of Courses where syllabus content has been changed (20% and more)

S. No.	Course Code	Name of the course	Percentage of Content changed	Page Number in which Details are Highlighted
1.	20BT31232	Programming with Python Lab	100	3
2.	20BT41532	Workshop in Open source Tools	66	6
3.	20BT51531	Linux Programming Lab	100	10
4.	20BT61531	Handheld computing: design and application Development Lab	100	12
5.	20BT71532	Data science using R Lab	100	14
Average % (A)			93.2	-
Total No. of Courses in the Program (T)			116	
No. of Courses where syllabus (more than 20% content) has been changed (N)			5	
Percentage of syllabus content change in the courses (C) = $(A \times N) / 100$			4.66	
Percentage of Syllabus Content changed in the Program (P) = $C / T \times 100$			4.0	


DEAN (Academics)
 DEAN (Academic)
 SREE VIDYANIKETHAN ENGINEERING COLLEGE
 Sree Sainath Nagar, A. RANGAMPET
 CHITTOOR (DT.)-517 102, A.P.


PRINCIPAL
 PRINCIPAL
 SREE VIDYANIKETHAN ENGINEERING COLLEGE
 (AUTONOMOUS)
 Sree Sainath Nagar, A. RANGAMPET
 Chittoor (Dist.) - 517 102, A.P., INDIA

II B. Tech. – I Semester
(20BT31232) PROGRAMMING WITH PYTHON LAB
(Skill Oriented Course)

(Common to CSSE, IT & CSBS)

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	1	2	2

PRE-REQUISITES: A course on Programming for Problem Solving.

COURSE DESCRIPTION: Hands on practice on Python programming fundamentals, control structures, Sequences, Modular programming, Data representation and Visualization.

COURSE OBJECTIVES:

- To demonstrate practical knowledge on python programming constructs.
- To develop solutions for societal problems using python programming.

COURSE OUTCOMES: After successful completion of the course, students will be able to:

CO1. Demonstrate the practical knowledge on python programming constructs.

CO2. Apply python programming constructs to provide solutions for logical and societal problems.

CO3. Use python modules for data representation and visualization.

CO4. Work independently or in teams to solve problems with effective communication.

LIST OF EXPERIMENTS:

Introduction to Python: Introduction - tokens, literals, identifiers, keywords, special symbols and operators, fundamental data types, expressions, type conversions, Input and output, Selection Statements, Iterative Statements.

- 1) a) Write a python program to perform the arithmetic operations on two integer numbers.
b) Write a python program to evaluate the following expressions by reading necessary inputs from the keyboard.
i) $ax^2 + bx + c$ ii) $ax^4 + bx^2 + c$ iii) $(ax - b) / (ax * b)$ iv) be^{kt}
- 2) a) Write a python program to swap two variable values without using a third variable.
b) Write a python program to read four integer values separated with commas and display the sum of those four numbers.
c) Write a python program to find roots of a Quadratic equation.
- 3) a) Write a python program that computes amount payable after discount from the price and quantity. The discount on amount is as follows:

- i) No discount when the amount less than or equal to 2000.
- ii) 5% discount when the amount exceeds 2000.
- iii) 10% discount when the amount exceeds 5000.
- iv) 15% discount when the amount exceeds 10000.

- b) Write a python program to find the largest among three numbers using nested if else.
- c) Write a python program to print the following pattern.

```

          1
        1 2 1
      1 2 3 2 1
    1 2 3 2 1
  1 2 3 2 1
1 2 3 2 1

```

- 4) a) Write a python program to check the given number is a Fibonacci number or not.
- b) Write a python program to find Armstrong numbers between $N1$ and $N2$, where $N1$ and $N2$ are two integer numbers and $N1 < N2$.
- c) Write a python program to sort n numbers without using a predefined function.

Sequences: Lists - operations, comprehensions, nested lists, tuples, nested tuples, strings - operations, string handling methods, string formatting, dictionaries - operations, dictionary methods, sorting elements using lambdas.

- 5) a) Write a python program to read n student details like name, roll number, branch and age. Sort the student details based on their names and display.
- b) Write a python program to delete duplicate strings from a list of strings. (Insertion order should maintain after deleting duplicate string).
- c) Write a python program to count the occurrences of each word in a given string.

- 6) Write a python program to implement the following conversions:

- i) Decimal to Binary
- ii) Binary to Decimal

- 7) a) Write a python program to read n number of student details into nested list and convert that as a nested dictionary.
- b) Write a python program to generate a dictionary that contains numbers between 1 and n in the form $(x, x+x)$.

Modular Programming: Introduction to functions, scope and lifetime, return statement, positional, keyword, default arguments and variable-length arguments, recursive functions, NumPy module.

- 8) a) Write a python function to perform the sum of integers that are sent to a function as variable length arguments.
- b) Write a python program to implement the following using recursion:

- i) GCD of two numbers
- ii) Towers of Hanoi problem.

- 9) a) Write a python program to perform arithmetic operations on NumPy arrays.
- b) Write a python program to perform following matrix operations using NumPy.

- i) Dot product
- ii) Matrix product
- iii) Determinant
- iv) Inverse

- 10) Write a python program that works as a scientific calculator using functions.

Data Representation and Visualization: *Pandas - data frame creation, reading data from CSV files, indexing and selecting data, dealing with rows and columns, Visualization - bar plots, histogram, Scatter Plot.*

- 11) a) Write a python program to create Pandas DataFrame using list of lists.
- b) Write a python program to load data from a CSV file into a Pandas DataFrame and perform basic operations on it.
- 12) a) Draw a Scatter Plot by considering an appropriate data set.
- b) Draw histograms by considering an appropriate data set.

REFERENCE BOOKS:

- 1. R. Nageswara Rao, *Core Python Programming*, 2nd Edition, Dreamtech Press, 2018.
- 2. R. G. Dromey, *How to solve it by Computer*, Pearson, 2006.
- 3. Reema Thareja, *Python Programming using Problem Solving Approach*, 1st Edition, Oxford University Press, 2017.
- 4. Charles Dierbach, *Introduction to Computer Science using Python: A Computational Problem-Solving Focus*, Wiley India, 2016.

II B. Tech. – II Semester

(20BT41532) WORKSHOP IN OPEN SOURCE TOOLS

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	-	3	1.5

PRE-REQUISITES: A course on “Programming for Problem Solving”, “Object Oriented Programming through Java”.

COURSE DESCRIPTION:

Configuration of Git; Initializing, Moving the local repository to the Git hub; I/O management system using Markdown and Jekyll themes; Quiz Game; Hacker Rank Problem; Phone Number Normalize; Building Images; Building PDFs; Building the Apps Hello Codi, Ball Bounce and Digital Doodle.

COURSE OUTCOMES:

On successful completion of this course, the students will be able to:

- CO1.** Write programs using Go programming Language constructs to solve basic problems.
- CO2.** Solve real world problems like Quiz Game, Hacker Rank Problems etc by applying command line arguments, strings.
- CO3.** Examine the Git Hub for configuring, initializing and moving local repositories using various Git commands.
- CO4.** Develop Web pages include I/O management system and Text Processing using branch, merge techniques and markdown.
- CO5.** Build simple mobile applications using MIT App inventor.
- CO6.** Work independently and communicate effectively in oral and written Forms.

LIST OF EXPERIMENTS:

I: Source Code Management Using Git and Git Hub:

- 1 Configuration of Git using gitconfig, gitlog and gitaliases commands.
- 2 Initializing the local repositories using Git commands: add, commit and status.
- 3 Moving the local repository to the Git hub using the command s remote, push and pull.
- 4 Creation of remote access to Github using SSH key
- 5 I/O management system with Github using branch and merge techniques
- 6 Text Processing in Github using various styles and formats, Creating Ordered/Unordered list, Adding videos/ pdfs to the markdown file.
- 7 Creation of personal portfolio site using Markdown and Jekyll themes for repositories.

II: Go Programming Language

- 1 Quiz Game: Create a program to run timed quizzes via the command line arguments.
- 2 Hacker Rank Problem: Code the solutions to a few string-related hacker rank problems. Ex: Prints the text of each line that appears more than once in the standard input, preceded by its count.
- 3 Phone Number Normalize: Write a program that will normalize an SQL table of phone numbers into a single format.
- 4 Building Images (png): Use the standard library to build a simple PNG bar chart for Creating images with Go code.
- 5 Building PDFs: Create any PDF in Go for generating invoice on set of purchased items.

III: Build Apps with MIT App Inventor:

- 1 Building the App “HelloCodi”: Select components to design app, Programming with the Blocks Editor, Playing the Sound, Packaging app.

- 2 Building the simple Game APP "BallBounce".
- 3 Building the Drawing App "DigitalDoodle".

REFERENCE BOOKS:

GIT Hub:

1. Scott chacon, Ben Straub, "**Pro Git**", Second Edition, APress open, 2014.

Go Programming Language:

1. Alan A. A. Donovan, Brain W. Kernighan "**The Go Programming Language**" First Edition, Addison-Wesley, New York, 2005.
2. <https://gophercises.com/>.

MIT App Inventor:

1. <http://appinventor.mit.edu/>

SOFTWARES/Tools used:

System Software: Git hub, Go Compiler, MIT App Inventor.

II B. Tech. – I Semester
(19BT31533) WORKSHOP IN COMPUTER SCIENCE AND SYSTEMS ENGINEERING

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
50	50	100	-	-	2	1

PRE-REQUISITES: A course on "Programming for Problem Solving"

COURSE DESCRIPTION:

Hands-on practice PC Hardware; Installation of Operating System; Software and Hardware Troubleshooting; Microsoft Office- Word and Excel; C Language- Operators, Expressions, Decision Making Statements, Looping Statements, Arrays and Functions.

COURSE OUTCOMES: *On successful completion of this course, the students will be able to:*

- CO1:** Devise solutions to the basic problems using C Language constructs.
- CO2:** Solve problems by applying functions, structures, dynamic memory allocation and pointers.
- CO3:** Develop, maintain and modify Web pages effectively using markdown.
- CO4:** Design personal portfolio in customized style by using git and Jekyll themes.
- CO5:** Build simple mobile applications using MIT App inventor.
- CO6:** Work independently and communicate effectively in oral and written forms.

LIST OF EXPERIMENTS:

I: C Programming:

- 1 a** Write a program to evaluate the following algebraic expressions after reading necessary values from keyword.
 - i) $(ax+b)/(ax-b)$
 - ii) $2.5\log x + \cos 32^\circ + |x^2 + y^2| + 2xy$
 - iii) $x^5 + 10x^4 + 8x^3 + 4x + 2$
 - iv) ae^{kt}
- b** Write a program to calculate commission for the input value of sales amount. Commission is calculated as per the following rules:
 - i) Commission is **NIL** for sales amount Rs. 5000.
 - ii) Commission is 2% for sales when sales amount is >Rs. 5000 and <= Rs. 10000.
 - iii) Commission is 5% for sales amount >Rs. 10000.
- 2 a** Write a program to find the grace marks for a student using switch. The user should enter the class obtained by the student and the number of subjects he has failed in. Use the following rules:
 - i. If the student gets first class and the number of subjects failed is >3, then no grace marks are awarded. If the number of subjects failed is less than or equal to '3' then the grace is 5 marks per subject.
 - ii. If the student gets second class and the number of subjects failed in is >2, then no grace marks are awarded. If the number of subjects failed in less than or equal to '3' then the grace is 4 marks per subject.
 - iii. If the student gets third class and the number of subjects failed in is >1, then no grace marks are awarded. If the number of subjects failed in is equal to '1' then the grace is 5 marks per subject
- b** Write a program to find the sum of individual digits of a positive integer using for loop
- 3 a** Write a program to generate all the prime numbers between 1 and N using while loop
- b** Write a program to generate Fibonacci sequence for N numbers using do-while loop.

- 4 **a** Write a program to perform the following: i) Addition of two matrices.
 ii) Multiplication of two matrices.
- b** Write a program to implement (i) Call by value (ii) Call by reference.
- 5 **a** Write a program to find factorial of a given number using recursion.
- b** Write a program that uses functions to perform the following operations:
 Write a program to determine whether the given string is palindrome or not.
- 6 Define a structure to store employee's data with the following specifications:
 Employee-Number, Employee-Name, Basic pay, Date of Joining
 i. Write a function to store 10 employee details.
 ii. Write a function to implement the following rules while revising the basic pay.
 If Basic pay \leq Rs.5000 then increase it by 15%.
 If Basic pay $>$ Rs.5000 and \leq Rs.25000 then it increase by 10%.
 If Basic pay $>$ Rs.25000 then there is no change in basic pay.
 Write a function to print the details of employees who have completed 20 years of
 service from the date of joining.
- 7 **a** Write a Program to calculate the sum of n numbers entered by the user using
 dynamic memory allocation functions.
- b** Write a Pointer Program to swap two numbers without using the 3rd variable.

II: Source Code Management Using Git and GitHub:

- 8 Installing Git, Configuring Git, Creating a Git repository, Creating and editing files,
 Adding files to Git repository, Making changes and tracking them, Synchronizing
 local Git repository with GitHub, Deleting and renaming files.
- 9 Markdown Syntax: Adding text of various styles and formats, Adding images along
 with text, Creating Ordered/Unordered list,
 Adding videos/ pdfs to the markdown file, adding links in the markdown file.
- 10 GitHub Pages: Creation of personal portfolio site- Creating a GitHub Page using
 Markdown and Jekyll themes for repositories.

III: Build Apps with MIT App Inventor:

- 11 Building the App "HelloCodi": Select components to design app, Programming with
 the Blocks Editor, Playing the Sound, Packaging app.
- 12 Building the simple Game APP "Ball Bounce".
- 13 Building the Drawing App "Digital Doodle".

REFERENCE BOOKS:

MIT App Inventor:

1. <http://appinventor.mit.edu/>

GIT Hub:

1. Scott chacon, Ben Straub, "**Pro Git**", Second Edition, APress open, 2014.

C LANGUAGE:

1. Behrouz A. Forouzan and Richard F. Gilberg, "**A Structured Programming Approach using C**," Third Edition, Cengage Learning, New Delhi, 2007.
2. Pradip Dey and Manas Ghosh, "**Programming in C**," Second Edition, Oxford University Press, New Delhi, 2007.

SOFTWARES/Tools used:

System Software: C Compiler/Code Blocks, MIT App Inventor, Git hub.

III B. Tech. – I Semester
(20BT51531) LINUX PROGRAMMING LAB

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	1	2	2

PRE-REQUISITES:

COURSE DESCRIPTION:

Hands on Practice with – Linux Commands; Shell Programs; System Calls; Environment Variables; Inter Process Communication; File System; I/O Libraries and Socket Programming.

COURSE OUTCOMES:

After successful completion of this course, the students will be able to:

- CO1:** Apply Linux programming constructs to implement modules for File and Process Management.
- CO2:** Analyze Inter-Process Communication techniques in Linux Environment.
- CO3:** Develop multithreaded applications for process communication in Client-Server environment.
- CO4:** Work independently or in teams to solve problems with effective communication.

List of Exercises:

1. Create two files source.txt and dest.txt using vi editor which contains some text and practice the following commands on those files. cat, tail, head, sort, nl, uniq, grep, egrep, fgrep, cut, paste, join, tee, comm, cmp, diff, cp, mv, ln, rm, unlink, tty, script, clear, date, cal, mkdir, rmdir, du, df, find, umask, ps, who, sed, netstat.
2. a. Write a shell script that takes a command line argument and reports on whether it is directory, a regular file, or any special file.
b. Write a shell script that accepts one or more file names as arguments and converts all of them to uppercase, provided they exist in the current directory.
3. a. Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
b. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
c. Write a shell script to display Fibonacci series up to given number.
4. a. Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.
b. Simulate cat command using C.
5. a. Simulate uniq command using C.
b. Simulate grep command using C.
6. Write a C program that takes one or more file or directory names as input and reports the following information on the file:
 - a. File type
 - b. Number of links
 - c. Read, write and execute permissions
 - d. Time of last access
 - e. Time of last modification(Note: Use stat/fstat system calls)
7. a. Write a C Program to display Environment variables.
b. Write a C Program to implement any two exec() functions.
c. Write a Program to create a Zombie Process by using fork() system call.

8. a. Write a C Program to store 20 random numbers in a file. Calculate and display the even random numbers sum and odd random numbers sum of the file. (Hint: Use I/O library file related functions).
b. Write a C Program to read "Name of the student, Roll Number, any three subject marks" in a string and display the "Name of the student, Roll Number and total marks" by using I/O library string related functions. (Hint: use sscanf, delimiters if necessary).
9. Implement the Following IPC Forms a) FIFO b) PIPE
10. Perform client and server socket Programming for exchanging of data Using System calls.
11. a. Write a program user.c, which extracts some user information from the password database (Hint: Use gethostname() function to obtain the network name of the host computer)
b. Write a program host.c, which extracts some host computer information (Hint: Use getuid() function to obtain the UID of the current user and use UID to obtain detailed password file information).
12. Write a client server program using FIFO in C language.

REFERENCE BOOKS/LABORATORY MANUALS:

1. Neil Matthew and Richard Stones, Beginning Linux Programming, Wiley Dreamtech, Fourth Edition, 2008.
2. Sumitabha Das, UNIX Concepts and Applications, Tata McGraw-Hill, Fourth Edition, 2008
3. Yashavant P. Kanetkar, UNIX Shell Programming, BPB Publications

SOFTWARE/Tools used:

- Linux Putty Application
- GCC Compiler

ADDITIONAL LEARNING RESOURCES:

- http://linuxcommand.org/lc3_learning_the_shell.php
- <https://tutorials.ubuntu.com/tutorial/command-line-for-beginners#0>
- <https://www.javatpoint.com/linux-tutorial>
- <http://www.freeos.com/guides/lsst/>

III B. Tech. – II Semester

(20BT61531) Handheld Computing: Design and Application Development Lab

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	1	2	2

PRE-REQUISITES:-

COURSE DESCRIPTION:

Setting up **Android Studio**, Programming with java and XML, Building **Android** Applications that uses GUI components and database, designing graphical primitives.

COURSE OUTCOMES:

CO1 Demonstrate the basic knowledge on basic requirements of mobile applications.

CO2 Analyze the challenges in mobile application design and development.

CO3 Create the designs for software development using Android SDK.

CO4 Design software applications with files and database connectivity.

Theory Component:

(10 Periods)

Android Studio IDE, User Interface development, Look and feel: to design layout and events, calculator application, database, storage, RSS Feed, GPS tracker, alert system, alarm

LIST OF EXPERIMENTS:

1. Develop an application that uses GUI components, Font and Colors.
2. Develop an application that uses Layout Managers and event listeners.
3. Develop a native calculator application.
4. Write an application that draws basic graphical primitives on the screen.
5. Develop an application that makes use of database.
6. Develop an application that makes use of RSS Feed.
7. Implement an application that implements Multi-threading.
8. Develop a native application that uses GPS location information.
9. Implement an application that writes data to the SD card.
10. Implement an application that creates an alert upon receiving a message.
11. Write a mobile application that creates alarm clock.

REFERENCE BOOKS

1. Jeff McWherter, Scott Gowell, "Professional Mobile Application Development paperback", Wiley India Private Limited, 2012.
2. Reto Meier, "Professional Android 2 Application Development", Wrox Wiley, 2010.
3. James Dovey and Ash Furrow, Beginning Objective C, Apress, 2012
4. Wei-Meng Lee, "Beginning iPhone SDK Programming with Objective-C, Wrox Wiley, 2010.
5. "App Programming Guide for iOS-Apple developer" - 2014 Apple Inc.

WEB RESOURCES

1. <https://www.theserverside.com/tutorial/Mobile-application-development-tutorial>
2. <https://www.toptal.com/android/developing-mobile-web-apps-when-why-and-how>
3. <http://developer.android.com/develop/index.html>
4. <https://www.tutorialspoint.com/ios/index.htm>

IV B. TECH. – I SEMESTER

(20BT71532) DATA SCIENCE USING R LAB

Int. Marks	Ext. Marks	Total Marks	L	T	P	C
30	70	100	-	1	2	2

PRE-REQUISITES: A course on Data warehousing and Data Mining “Machine Learning”, “Data Visualization Techniques”

COURSE DESCRIPTION:

R Shell script; Descriptive statistics Reading and Writing different types of data sets, Visualization, correlation and covariance, Single and Multiple regression Models, Classification and Clustering Models.

COURSE OUTCOMES:

After successful completion of this course, the students will be able to:

- CO1.** Design R Scripts for handling different applications and finding descriptive statistics on data sets.
- CO2.** Develop Scripting programs for handling different types of data sets to perform read and write operations
- CO3.** Apply statistical distributions to visualize the large data and to find outliers.
- CO4.** Analyze the predictable data by applying single and Multiple Linear regression models.
- CO5.** Evaluate the performance of different machine learning techniques for a real world problem.
- CO6.** Work independently or communicate effectively in oral and written forms.

LIST OF EXPERIMENTS:

1. R AS CALCULATOR APPLICATION

Write an R script, to create R objects for calculator application and save in a specified location in disk.

2. DESCRIPTIVE STATISTICS IN R

- a. Write an R script to find basic descriptive statistics using summary, str, quartile function on mtcars & cars datasets.
- b. Write an R script to find subset of dataset by using subset (), aggregate () functions on iris dataset.

3. READING AND WRITING DIFFERENT TYPES OF DATASETS

- a. Reading different types of data sets (.txt, .csv) from Web and disk and writing in file in specific disk location.
- b. Reading Excel data sheet in R.
- c. Reading XML dataset in R

4. VISUALIZATIONS

- 1. Find the data distributions using box and scatter plot.
- 2. Find the outliers using plot
- 3. Plot the histogram, bar chart and pie chart on sample data.

5. CORRELATION AND COVARIANCE

- a. Find the correlation matrix.
- b. Plot the correlation plot on dataset and visualize giving an overview of relationships among data on iris data.
- c. Analysis of covariance: variance (ANOVA), if data have categorical variables on iris data.

6. REGRESSION MODEL

Import a data from web storage. Name the dataset and implement Logistic Regression to find out relation between variables that are affecting the admission of a student in a institute based on his or her GRE score, GPA obtained and rank of the student. Also check the model is fit or not. Require (foreign), require (MASS).

7. MULTIPLE REGRESSION MODEL

Apply multiple regressions on any dataset if data have a continuous Independent variable.

8. REGRESSION MODEL FOR PREDICTION

Apply regression Model techniques to predict the data on the given dataset.

9. CLASSIFICATION MODEL

- a. Install relevant package for classification.
- b. Choose classifier for classification problem.
- c. Evaluate the performance of classifier

10. CLUSTERING MODEL

- a. Clustering algorithms for unsupervised classification.
- b. Plot the cluster data using R visualizations

11. RECOMMENDER SYSTEMS

Implementing a simple Recommender System on user based rating and testing the same.

REFERENCE BOOKS:

1. Yanchang Zhao, "R and Data Mining: Examples and Case Studies", Elsevier, 1st Edition, 2012.

WEB REFERENCES:

1. <http://www.r-bloggers.com/how-to-perform-a-logistic-regression-in-r/>
2. <http://www.ats.ucla.edu/stat/r/dae/rreg.htm>
3. <http://www.coastal.edu/kingw/statistics/R-tutorials/logistic.html>
4. <http://www.ats.ucla.edu/stat/r/data/binary.csv>

SOFTWARES/Tools used:

1. R Studio Software