Third Year B.C.A. (Under Science) Semester V

Course Code: BCA501 **Course Title: Java Programming**

Total Contact Hours: 48 hrs. **Total Credits: 04** **Total Marks: 100**
(60 Lectures)

Teaching Scheme: Theory- 05 Lect./ Week

Course Objectives:

The syllabus aims in equipping students with

- To understand fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries, etc.
- To handle abnormal termination of a program using exception handling

To use the Java SDK environment to create, debug and run simple Java program

Unit 1: Introduction to Java	**04**
Basics of Programming Language • History and Features of Java • JDK,JRE,JIT and JVM • Naming Convention • Simple java program • Java IDE –Eclipse/ NetBeans (Note: For Lab Demonstration) **Introduction to Java** Data Types Variable: final, static, abstract • Types of Comments • Array: 1D, 2D, Dynamic array using Vector • Accepting input using Command line argument • Accepting input from console (Using BufferedReader and Scanner class)	

Unit 2: Usage of Objects and Classes	**04**
• Defining Your Own Classes • Access Specifiers (public, protected, private, default/friendly) • Array of Objects • Constructors, Overloading Constructors and use of ‘this’ Keyword • Predefined classes ➢ String class (Basic Functions)	

➢ StringBuffer class ➢ Wrapper class • Inner classes, Nested classes, local classes, Anonymous classes(Anonymous object) • Introduction to Packages : Creation, Access and use • Garbage Collection (finalize() Method)	

Unit 3: Inheritance and Interface	**03**
• Inheritance Basics (extends Keyword) • Types of Inheritance • use of 'super' Keyword • Usage of final keyword related to method and class • Usage of abstract class and abstract methods • Interface: Defining and Implementing Interfaces • Runtime polymorphism using interface	

Unit 4: Collection	**07**
• Collection interface ➢ Collection framework ➢ Collection interfaces & classes-ArrayList, LinkedList, HashSet, TreeSet • Iterator and Enumaration, Hash Table. • Vector.	

Unit 5: Exception Handling and I/O	**10**

• Exception handling fundamentals • Exception types • Exception class ➢ Checked exception ➢ Unchecked exception • Creating user defined exception • Uncaught exceptions • Assertions • Introduction to Java.io package • Byte streams • Character streams • File IO basics • Object serialization – Reader and Writer	

Unit 6: Swing , Applet programming	**09**
• MVC(Model View Controller) Architecture • Swing components : JFrame, JPanel, JButton, JcheckBox, JtextField, JRadioButton, JLabel, JList, JDialog, JFileChooser, JColorChooser,JMenu • Applet fundamentals, Applet lifecycle, Creating and running applets • Applets: Event Handling using applets.	

Unit 7: Database Programming	**8**
• Introduction to JDBC: Architecture (2-tier, 3-tier) • JDBC Drivers • Connectivity with PostgreSQL: basic steps • JDBC statement: Statement, PreparedStatement, CallableStatement • JDBC ResultSet and types • JDBC Metadata – ResultSetMetaData, DatabaseMetaData	

Unit 8: Servlets	08
• Introduction to Servlet and Servlet types	

• Lifecycle of servlet • Handing HTTPRequest and HTTPResponse • HttpServlet: ➢ Reading form data from servlet ➢ Servlet - Database communication • Session tracking –User Authorization, URL Rewriting, Hidden Form fields, Cookies and HttpSession	

Unit 9: Java Server Pages (JSP)	**7**
• Introduction to JSP • Life cycle of JSP • Implicit Objects • Scripting elements –Declarations, Expressions, Scriplets, Comments • JSP Directives – Page Directive, include directive • Basic JSP program • Mixing Scriplets and HTML • Example of forwarding contents from database to servlet, servlet to JSP and displaying it using JSP scriplet tag	

Reference Books:

1. Complete reference Java by Herbert Schildt(5th edition)
2. Java 2 Programming Black Book, Steven Horlzner
3. Programming with java, a Primer, 4th edition, By e balgurusamy
4. Core Java Volume I- Fundamentals, 8th edition, Cay S Horstmann,Gary Cornell, Prentice Hall, Sun MicroSystem Press
5. Core Java Volume II- Advance Features, 8th edition, Cay S Horstmann, Gary Cornell, Prentice Hall, Sun MicroSystem Press

Third Year B.C.A. (Under Science) Semester V

Course Code: BCA 502 **Course Title: Advanced Web Technology**
Total Contact Hours: 48hrs.(60 lectures) **Total Credits: 04**
Total Marks: 100 **Teaching Scheme: Theory- 05 Lect. /Week**

Objectives -:

1. To know & understand concepts of internet programming.

Unit No	Contents	No Of Lectures
1	**Introduction to Object Oriented Programming in PHP** 1.1 Classes 1.2 Objects 1.3 Encapsulation 1.4 Constructor and Destructor 1.5 Inheritance 1.6 Interfaces 1.7 Introspection	10
2	**Web Techniques** 2.1 Super global Variables 2.2 Server information 2.3 Sticky forms 2.4 File Uploads 2.5 Setting response headers 2.6 Maintaining state 2.6.1. Session and Cookies	12
3	**Files and Directories** 3.1 Working with files and directories 3.2 Opening and Closing 3.3 Getting information about file 3.4 Reading and writing characters in file 3.5 Rename and delete files 3.6 Random access to file data 3.7 Getting information on file 3.8 Ownership and permissions	10
4	**Databases(Postgresql)** 4.1 Using PHP to access/insert/update/delete a database tables 4.2 Relational databases and SQL 4.3 Introduction to PEAR DB basics (No assignments) 4.4 Advanced database techniques 4.5 Simple applications	9
	XML 5.1 What is XML?	

5	5.2 XML document Structure 5.3 PHP and XML 5.4 XML parser 5.5 The document object model 5.6 The simple XML extension 5.7 Changing a value with simple XML	8
6	**Ajax** 6.1 Understanding java scripts for AJAX 6.2 AJAX web application model 6.3 AJAX –PHP framework 6.4 Performing AJAX validation 6.5 Handling XML data using php and AJAX 6.6 Connecting database using php and AJAX	8
7	**Introduction to Web Services** 7.1. SOAP 7.2. WSDL 7.3. Application of web services	3

Reference Books : -

1. Complete HTML- Thomas Powell
2. HTML and JavaScript–Ivan Bayross
3. Programming PHP By Rasmus Lerdorfand Kevin Tatroe, O'Reilly publication
4. Beginning PHP5, Wrox publication
5. PHP for Beginners, SPD publication

Third Year B.C.A. (Under Science) Semester V

Course Code: BCA 503 **Course Title: Software Quality Assurance**

Total Contact Hours: 48 hrs. **Total Credits: 04** **Total Marks: 100**
(60 Lectures)

Teaching Scheme: Theory- 05 Lect./ Week

Pre-requisites(if any) :

1. Basic concepts of Software Engineering

Course Objectives:

1. To **u**nderstand the basic of quality software and quality factors.
2. To understand software quality architecture and component.
3. To understand software project life cycle, infrastructure and software quality standards**.**

Unit No.	**Contents**	**No. of Lectures**
Unit 1	**1. Introduction to Software Quality.** 1.1. Uniqueness of software quality assurance 1.2. Software, Software errors, Faults and Failures 1.3. Classification of the causes of software errors 1.4. Software quality, Software quality assurance and software engineering	06
Unit 2	**2. Software Quality Architecture and Components** 2.1. The need for comprehensive software quality requirements 2.2. Classifications of software requirements into software quality factors 2.2.1. Product Operation 2.2.2. Product Revision 2.2.3. Product Transition 2.3. Parties interested in the definition of quality requirements. 2.4. SQA architecture 2.5. Software Quality Components 2.5.1. Pre-project components 2.5.2. Software project life cycle components 2.5.3. Infrastructure components for error prevention and improvement 2.6. Management SQA components	10
Unit 3	**3. Project Life Cycle.** 3.1. Classic and other software development methodologies 3.2. Factors affecting intensity of quality assurance activities in the development process 3.3. Verification, validation and qualification 3.4. A model for SQA defect removal effectiveness and cost	14

	3.5. Demonstration of CASE study and CASE tools 3.5.1. What is a CASE tool? 3.5.2. The contribution of CASE tools to software product quality 3.5.3. The contribution of CASE tools to software maintenance quality 3.5.4. The contribution of CASE tools to improved project management	
Unit 4	**4. Software Quality Infrastructure Components** 4.1. Procedures and work instructions – 4.1.1. Need 4.1.2. Procedures manuals 4.1.3. work instruction manuals 4.1.4. Procedures and work instructions: preparation, implementation and updating 4.2. Supporting Quality devices 4.2.1. Templates 4.2.2. Checklists 4.3. Configuration management - 4.3.1. Software configuration, its items and its management 4.3.2. Software configuration management – tasks and organization 4.3.3. Software change control 4.3.4. Release of software configuration versions 4.3.5. Provision of SCM information services 4.3.6. Software configuration management audits 4.3.7. Computerized tools for managing software configuration	10
Unit 5	**5. Software quality metrics** 5.1. Objectives of quality measurement 5.2. Classification of software quality metrics 5.3. Process metrics 5.4. Product metrics 5.5. Implementation of software quality metrics 5.6. Limitations of software metrics	10
Unit 6	**6. Software Quality Standards, certification and assessment** 6.1. Quality management standards 6.1.1. The scope of quality management standards 6.1.2. ISO 9001 and ISO 9000-3 6.1.3. Certification according to ISO 9000-3 6.1.4. Capability Maturity Models – CMM and CMMI assessment methodology 6.1.5. The Bootstrap methodology 6.1.6. The SPICE project and the ISO/IEC 15504 software process assessment standard 6.2. Project process standards 6.2.1. Structure and content of IEEE software engineering standards	10

	6.2.2. IEEE/EIA Std 12207 – software life cycle processes 6.2.3. IEEE Std 1012 – verification and validation 6.2.4. IEEE Std 1028 – reviews	

Reference Books:

1. Software Quality Assurance by Daniel Galin, Pearson Publication, 2009.
2. Software testing and Quality Assurance Theory and Practice by Kshirasagar Naik and Priyadarshi Tripathy, Wiley Publication.
3. Software Engineering A Practitioner’s Approach Sixth Edition by Roger S. Pressman, McGraw Hill Publication
4. Metrics and Models in Software Quality Engineerning, By Stephen H. Kan, Pearson Publication

Third Year B.C.A. (Under Science) Semester V

Course Code: BCA504 **Course Title: Operating Systems**

Total Contact Hours: 48 hrs. (60 Lectures) **Total Credits: 04** **Total Marks: 100**

Teaching Scheme: Theory- 05 Lect./ Week

Pre-requisites : Knowledge of fundamentals of Computer Organization

Course Objectives:

1. To understand the objectives, structure and functions of operating system
2. To learn about concept of processes, threads and its scheduling algorithms
3. To understand design issues in process synchronization and deadlock management
4. To study various memory management schemes
5. To learn about concept file and I/O management in detail.

Unit No.	Content	No. of Lectures
Unit 1	**1. Introduction to Operating System Concepts** 1.1 Operating System Objectives and Functions - Definition of Operating System ,Role and Objectives of Operating System, Operating System as a User View and System View 1.2 Evolution Of Operating Systems - Batch Operating System, Multi-Programming Operating System ,Time-Sharing Operating System, Desktop Operating Systems, Real-Time Operating System, Distributed Operating System, Parallel Systems, Multimedia Systems, Handheld Systems 1.3 Computer System Architecture - Single-Processor Systems, Multi-Processor Systems, Clustered Systems 1.4 Operating System Operations- Dual-Mode And Multimode Operation, Timer 1.5 Operating System as Resource Management- Process Management, Memory Management, Storage Management(File system ,Mass storage ,Caching I/O systems), Protection And Security 1.6 Computing Environment-Traditional Computing, Client Server Computing, Peer To Peer Computing, Virtualization, Cloud Computing	**06**
Unit 2	**2. System structure** 2.1 Operating System Services 2.2 System Calls Concepts 2.3 Types of system Calls- Process Control, File Management, Device Management, Information Maintenance, Communication, Protection 2.4 System Programs	**03**

	2.5 System Boot	
Unit 3	**3. Process and Thread Management** 3.1 Process Concept – Process , Process Model , Process Control Block 3.2 Operations on Process – Process creation , Process Termination 3.3 Process Scheduling - Scheduling queues, Schedulers, Context switch 3.4 Inter Process Communication –Cooperating Process, Shared Memory Systems, Message Passing Systems 3.5 Overview of Threads 3.6 Concept of Multithreaded Programming and Multicore Programming 3.7 Types of threads – User and Kernel 3.8 Multithreading Models – Many to One , One to Many, Many to Many	**06**
Unit 4	4 **Process Scheduling** 4.1 Basic Concept – CPU-I/O burst cycle, CPU scheduler, Preemptive scheduling, Dispatcher 4.2 Scheduling Criteria 4.3 Scheduling Algorithms – FCFS, SJF, Priority scheduling, Round-robin scheduling, Multiple queue scheduling, Multilevel feedback queue scheduling	**06**
Unit 5	5 **Process Synchronization** 5.1 Background – Problems with Concurrency , Race Condition 5.2 Critical Section Problem – Peterson’s Solution(for two process) 5.3 Semaphores: Usage, Implementation 5.4 Classic Problems of Synchronization – Producer Consumer problem, Reader Writer problem, Dining Philosopher Problem	**05**
Unit 6	6 **Deadlocks** 6.1 System Model 6.2 Deadlock Characterization – Necessary Conditions, Resource Allocation Graph 6.3 Deadlock Prevention 6.4 Deadlock Avoidance - Safe State, Resource Allocation Graph Algorithm, Banker’s Algorithm 6.5 Deadlock Detection 6.6 Recovery From Deadlock – Process Termination, Resource Preemption	**08**
Unit 7	7 **Memory Management** 7.1 Background – Basic Hardware, Address Binding, Logical Versus Physical Address Space, Dynamic Loading, Dynamic Linking and Shared Libraries, Overlays 7.2 Swapping 7.3 Contiguous Memory Allocation – Memory Mapping and Protection, Memory Allocation, Fragmentation 7.4 Paging – Basic Method, Hardware Support, Protection, Shared	**12**

	Pages 7.5 Segmentation – Basic Concept, Hardware 7.6 Virtual Memory Management – Background, Demand Paging 7.7 Page Replacement Algorithms – FIFO, OPT, LRU, Second Chance Page Replacement, LFU, MFU. 7.8 Thrashing – Cause of thrashing , Working-set Model	
Unit 8	8 **File System** 8.1 File concept 8.2 Access Methods – Sequential, Direct, Other access methods 8.3 Directory and Disk Structure – Storage structure, Directory overview, Single level directory, Two level directory, Tree structure directory, Acyclic graph directory, General graph directory 8.4 Allocation Methods – Contiguous allocation, Linked allocation, Indexed allocation 8.5 Free Space Management – Bit vector, Linked list, Grouping, Counting, Space maps	**07**
Unit 9	9 **I/O Systems and Disk Management** 9.1 I/O Hardware - polling, interrupts, DMA 9.2 Application I/O Interface - block and character devices, network devices, clocks and timers, blocking and non blocking I/O 9.3 Kernel I/O subsystems - (I/O scheduling, buffering, caching, spooling and device reservation, error handling) 9.4 Disk Structure 9.5 Disk Scheduling – Disk Performance Parameters, Scheduling algorithms(FCFS, SSTF, SCAN,C-SCAN,LOOK,C-LOOK)	**07**

Reference Books:

1. “Operating System Concepts”, 9th Edition ,by Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, John Wiley & Sons (ASIA) Pvt. Ltd, 2013.
2. “Operating Systems: Internals and Design Principles”, 7th Edition, by William Stallings, Prentice Hall, 2011
3. “Modern Operating Systems”, 4th Edition, by Andrew S. Tanenbaum, Prentice Hall of India Pvt. Ltd, 2014.
4. “Operating Systems : Principles and Design” – Pabitra Pal Choudhary (PHI Learning Private Limited)
5. “An Introduction to Operating Systems, Concepts and Practice” by Pramod Chandra P. Bhatt , PHI, 2010
6. “Operating Systems: A Concept -based Approach”, 2nd Edition by D M Dhamdhere, Tata McGraw -Hill Education, 2007.

Websites:

https://www. nptel.ac.in

http://www.scs.stanford.edu/17wi-cs140/

http://cnds.eecs.jacobs-university.de/courses/os-2016/slides.pdf

https://courses.cs.vt.edu/csonline/OS/Lessons/index.html

Third Year B.C.A. (Science) Semester V

(To be implemented from Academic year 2017-18)

Course Code: BCA-505 **Course Title: LAB I (Core Java)**

Total Contact Hours: 48 hrs. **Total Credits: 04**

Total Marks: 100

Note that these are only sample assignments. Teachers may conduct practicals by preparing similar types of examples

Java Sample Programs

1. Java Program to Print an Integer (Entered by the User)
2. Java Program to Add Two Integers
3. Java Program to Multiply two Floating Point Numbers
4. Java Program to Find ASCII Value of a character
5. Java Program to Compute Quotient and Remainder
6. Java Program to Swap Two Numbers
7. Java Program to Check Whether a Number is Even or Odd
8. Java Program to Check Whether an Alphabet is Vowel or Consonant
9. Java Program to Find the Largest Among Three Numbers
10. Java Program to Find all Roots of a Quadratic Equation
11. Java Program to Check Leap Year
12. Java Program to Check Whether a Number is Positive or Negative
13. Java Program to Check Whether a Character is Alphabet or Not
14. Java Program to Calculate the Sum of Natural Numbers
15. Java Program to Find Factorial of a Number
16. Java Program to Generate Multiplication Table
17. Java Program to Display Fibonacci Series
18. Java Program to Find GCD of two Numbers
19. Java Program to Find LCM of two Numbers

20. Java Program to Display Characters from A to Z using loop

21. Java Program to Count Number of Digits in an Integer

22. Java Program to Reverse a Number

23. Java Program to Calculate the Power of a Number

24. Java Program to Check Whether a Number is Palindrome or Not

25. Java Program to Check Whether a Number is Prime or Not

26. Java Program to Display Prime Numbers Between Two Intervals

27. Java Program to Check Armstrong Number

28. Java Program to Display Armstrong Number Between Two Intervals

29. Java Program to Display Prime Numbers Between Intervals Using Function

30. Java Program to Display Armstrong Numbers Between Intervals Using Function

31. Java Program to Display Factors of a Number

32. Java Program to Make a Simple Calculator Using switch...case

33. Java Program to Check Whether a Number can be Expressed as Sum of Two Prime Numbers

34. Java Program to Find the Sum of Natural Numbers using Recursion

35. Java Program to Find Factorial of a Number Using Recursion

36. Java Program to Find G.C.D Using Recursion

37. Java Program to Convert Binary Number to Decimal and vice-versa

38. Java Program to Convert Octal Number to Decimal and vice-versa

39. Java Program to Convert Binary Number to Octal and vice-versa

40. Java Program to Reverse a Sentence Using Recursion

41. Java Program to calculate the power using recursion

42. Java Program to Calculate Average Using Arrays

43. Java Program to Find Largest Element of an Array

44. Java Program to Calculate Standard Deviation

45. Java Program to Add Two Matrix Using Multi-dimensional Arrays

46. Java Program to Multiply to Matrix Using Multi-dimensional Arrays

47. Java Program to Multiply two Matrices by Passing Matrix to a Function

48. Java Program to Find Transpose of a Matrix

49. Java Program to Find the Frequency of Character in a String

50. Java Program to Count the Number of Vowels and Consonants in a Sentence

51. Java Program to Sort Elements in Lexicographical Order (Dictionary Order)

52. Java Program to Add Two Complex Numbers by Passing Class to a Function

53. Java Program to Calculate Difference Between Two Time Periods

54. Java Code To Create Pyramid and Pattern

55. Java Program to Remove All Whitespaces from a String

56. Java Program to Print an Array

57. Java Program to Convert String to Date

58. Java Program to Round a Number to n Decimal Places

59. Java Program to Concatenate Two Arrays

60. Java Program to Convert Character to String and Vice-Versa

61. Java Program to Check if An Array Contains a Given Value

62. Java Program to Check if a String is Empty or Null

63. Java Program to Get Current Date/TIme

64. Java Program to Convert Milliseconds to Minutes and Seconds

65. Java Program to Add Two Dates

66. Java Program to Join Two Lists

67. Java Program to Convert List (ArrayList) to Array and Vice-Versa

68. Java Program to Get Current Working Directory

69. Java Program to Convert Map (HashMap) to List

70. Java Program to Convert Array to Set (HashSet) and Vice-Versa

71. Java Program to Convert Byte Array to Hexadecimal

72. Java Program to Create String from Contents of a File

73. Java Program to Append Text to an Existing File

74. Java Program to Convert a Stack Trace to a String

75. Java Program to Convert File to byte array and Vice-Versa

76. Java Program to Convert InputStream to String

77. Java Program to Convert OutputStream to String

78. Java Program to Lookup enum by String value

79. Java Program to Compare Strings

80. Java Program to Sort a Map By Values

81. Java Program to Sort ArrayList of Custom Objects By Property

82. Java Program to Check if a String is Numeric

Third Year B.C.A. (Science) Semester V
(To be implemented from Academic year 2017-18)

Course Code: BCA-506 **Course Title: LAB II (Adv. Web Technology)**

Total Contact Hours: 48 hrs. **Total Credits: 04**

Total Marks: 100

Note that these are only sample assignments. Teachers may conduct practicals by preparing similar types of examples

PHP Slips for T Y BCA

1) Write class declarations and member function definitions for an **employee**(code, name, designation). Design derived classes as **emp_account**(account_no, joining_date) from employee and **emp_sal**(basic_pay, earnings, deduction) from emp_account.
Write a PHP Script to create 5 objects (pass details using **__**Construct () constructor) and Display details Employees who having Maximum and Minimum Salary.

2) Define an interface which has methods area(), volume(). Define constant PI. Create a class cylinder which implements this interface and calculate area and volume. (Use define())

3) Derive a class square from class Rectangle. Create one more class circle. Create an interface with only one method called area(). Implement this interface in all the classes. Include appropriate data members and constructors in all classes. Write a program to accept details of a square, circle and rectangle and display the area.

4) Create an abstract class Shape with methods calc_area( ) and calc_volume( ). Derive three classes Sphere(radius) , Cone(radius, height) and Cylinder(radius, height), Calculate area and volume of all. (Use Method overriding).

5) Define a class Employee having private members – id, name, department, salary. Define parameterized constructors. Create a subclass called "Manager" with private member bonus. Create 6 objects of the Manager class and display the details of the manager having the maximum total salary (salary + bonus).

6) Write a PHP Script to create a super class **Vehicle** having members Company and price. Derive 2 different classes LightMotorVehicle (members – mileage) and HeavyMotorVehicle (members – capacity-in-tons). Define 5 Object of each subclass and display details in table format.

7) Write PHP script for the following: Define Class declarations and member function definitions for Student(rollno, name, academic_year). Design derived classes as Internal(marks, total), External(marks, total). Perform the following operations and show the results: Accept the details from the user and Show the result along with all details and total marks as addition of marks of Internal and External.

8) Write a script to keep track of number of times the web page has been accessed(use $_COOKIE).

9) Create a login form with a username and password. Once the user logs in, the second form should be displayed to accept user details (name, city, phoneno). If the user doesn't enter information within a specified time limit, expire his session and give a warning otherwise Display Details($_SESSION).

10) Create a form to accept student information (name, class, address). Once the student information is accepted, accept marks in next form (Java, PHP, ST, IT, pract1, and project). Display the mark sheet for the student in the next form containing name, class, marks of the subject, total and percentage(Use $_COOKIE).

11) Write a program to create a shopping mall. User must be allowed to do purchase from three pages. Each page should have a page total. The fourth page should display a bill, which consists of a page total of whatever the purchase has been done and print the total. (Use $_SESSION).

12) Create a form to accept customer information (name, address, ph-no) (use Java Script to validate fields).Once the customer information is accepted, accept product information in the next form(Product name, qty, rate). Display the bill for the customer in the next form. Bill should contain the customer information and the information of the products entered.

13) Write a PHP script to accept username and password. If in the first three chances, username and password entered is correct, then display second form with well come massage, otherwise display error message.

14) Create student registration form and display details in the next page. (Use sticky form concept).

15) Write a PHP Script to display Server information in table format (Use $_SERVER).

16) Write a PHP Script to Upload the file and display its information.(use $_FILES).

17) Write a PHP program to accept username and password from the user. Validate it against the login table in the database. If there is a mismatch between username and password, then, display the error message as ―invalid user name and password‖, else display the message as ―Login successful‖ on the browser.

18) Write a PHP program to implement Create, Read, Update and Display operations on Employee table with attributes(empno, empname, dateOfJoin, address, salary). **(Use Radio Buttons)**

19) Consider the following relational database:

    Project (Pgroupno, ProjectTitle)
    Student (Seat no, Name, Class, Pgroupno)

Write a PHP script to accept project title and display list of students those who are working in a particular project.

20) Consider the following entities and their relationships

Emp (emp_no,emp_name,address,phone,salary)
Dept (dept_no,dept_name,location)

Emp-Dept are related with one-many relationship
Create a RDB in 3NF for the above and solve following
Using above database write a PHP script which will

a) Insert employee records in table .
b) Print a salary statement in the format given below, for a given department. (Accept department name from the user).

Maximum Salary	Minimum Salary	Sum Salary

21) Consider the following entities and their relationships

Doctor (doc_no, doc_name, address, city, area)
Hospital (hosp_no, hosp_name, hosp_city)

Doctor and Hospital are related with many-many relationship
Create a RDB in 3 NF for the above and solve following

Using above database, write a PHP script which accepts hospital name and print information about doctors visiting / working in that hospital in tabular format.

22) Consider the following entities and their relationships

Movie (movie_no, movie_name, release_year)
Actor (actor_no, name)

Relationship between movie and actor is many – many with attribute rate in Rs. Create a RDB in 3 NF for the above and solve following Using above database, write PHP scripts for the following:

(Hint: Create HTML form having two radio buttons)

a) Accept actor name and display the names of the movies in which he has acted.
b) Insert new movie information.

23) Consider the following entities and their relationships

BillMaster(billno, custname, billdate)
BillDetails(itemname, qty, rate, discount)

BillMaster and BillDetails are related with one-to-many relationship.
Create a RDB in 3 NF for the above and solve following
Write PHP script to print the bill in following format Accept the Bill number from user.

BillNo : **BillDate :**

Customer Name :

SrNo	**Particular**	**Quantity**	**Rate**	**Discount**	**Total**

Gross Amount :

24) Write a script to create XML file named “Rajashree.xml”

The element details of “Rajashree .xml” are:

```
<Rajashree Productions>
        <Movie>
            <MovieName>……….</MovieName>
           <Actor Name>…………</ActorName>
          <ReleaseYear>…………</ReleaseYear>
         </Movie>
</RajashreeProductions>
```

Store details of at least 5 movies which got released during 1990-2015.

25) Write a PHP script to generate an XML in the following format

```
<?xml version = “1.0” ?>
<BookStore>
          <Books>
                    <PHP>
                              <title>Programming PHP</title>
                              <publication>O’RELLY</publication>
                    </PHP>
                    <PHP>
                              <title>Beginners PHP</title>
                              <publication>WROX</publication>
                    </PHP>
          </Books>
</BookStore>
```

26) Create a XML file which gives details of books available in “ ABC Bookstore” from following categories.

1) Technical
2) Cooking
3) Yoga

and elements in each category are in the following format

```
<Book>
               <Book_PubYear>--------------</Book_PubYear>
               <Book_Title>      --------------</Book_Title>
               <Book_Author>   ---------------</Book_Author>
</Book>
```

Save the file as “Book.xml”

Create an application that reads “Book.xml” file into simple XML object. Display attributes and elements.

(Hint: Use simple_xml_load_file() function)

27) Write a script to solve following questions (Use “Book.xml” file)
a) Create a DomDocument object and load this XML file.
b) Get the output of this Document to the browser
c) Save this [. XML] document in another format i.e. in [.doc]
d) Write a XML program to print the names of the books available in “Book.xml” file.

28) Write a script to create “cricket.xml” file with multiple elements as shown below:

```
<CricketTeam>
    <Team country=”India”>
            <player>____</player>
            <runs>______</runs>
            <wicket>____</wicket>
    </Team>
</CricketTeam>
```

Write a script to add multiple elements in “cricket.xml” file of category, country=”Australia”.

29) Write a PHP script to accept an .XML file which should comprise the following:

```
<cricket>
        <player>abc</player>
        <runs>1000</runs>
        <wickets>50</wickets>
        <noofnotout>10</noofnotout>
</cricket>
```

For at least 5 players. Display the details of players who have scored more than 1000 runs and at least 50 wickets.

30) Link this “Rajashree.xml” file to the CSS style sheet and get well formatted output as given below.

a)MovieName :
Color: black,
Font-family: Copperplate Gothic Light;
Font Size: 16 pts;
Font:Bold;

b)ActorName and ReleaseYear:
Color: Red;
Font-family: Bodoni MT;
Font Size: 12 pts;
Font: Bold

31) Write a php script using AJAX concept, to give Hint to user when he/she type city name in the text field.

32) Write a PHP script using AJAX concept, to check user name and password are valid or Invalid (use database to store user name and password).

33) Write a PHP script using AJAX concept, to develop user-friendly and interactive search engine.

Third Year B.C.A. (Under Science) Semester V

Course Code: BCA 507 **Course Title: Soft Computing**

Total Contact Hours: 24 hrs. **Total Credits: 02** **Total Marks: 50**
(30 Lectures)

Teaching Scheme: Theory- 03 Lect./ Week

Pre-requisites :

Knowledge of Set Theory

Course Objectives:

1. To learn the concept of soft computing.

2. Understand different soft computing techniques like Genetic Algorithms, Fuzzy Logic , Neural Networks and their combination.

Unit No.	**Contents**	**No. of Lectures**
Unit 1	Introduction to Soft Computing 1.1 History of Soft Computing 1.2 Brief Introduction to Neural Networks, Genetic Algorithms and Fuzzy Systems 1.3 Applications of Soft Computing	**04**
Unit 2	Fundamentals of Fuzzy Systems 2.1 Fuzzy sets: Basic Definition and Terminology 2.2 Member Functions 2.3 Fuzzy Logic and Relations 2.4 Extension Principle and its problems	**09**
Unit 3	Fundamentals of Evolutionary Computing 3.1 Evolutionary Algorithms 3.2 Encoding 3.3 Operators of genetic Algorithms and its problems	**07**
Unit 4	Fundamental of Neural Network 4.1 Introduction 4.2 Model of Artificial Neuron 4.3 Architectures 4.4 Learning Methods (Supervised and Unsupervised) 4.5 Perceptron and Back-propagation	**10**
		30

<u>Reference Books:</u>

References

1.Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis & Applications, S. Rajasekaran, G. A. Vijayalakshami, PHI.

2.Chin Teng Lin, C. S. George Lee, Neuro-Fuzzy Systems, PHI

3. Timothy Ross,Fuzzy Logic with Engineering Applications, TMH
4 .Introduction to soft Computing ,Eva Volna ,1st Edition ,ISBN 978-87-403-0573-09
5. Kishan Mehrotra,Elements of Artificial Neural Network, MIT Press
6.E. Goldberg,Genetic Algorithms: Search and Optimization, Addision-Wesley
7. S.N. Sivanandan and S.N. Deepa, Principles of Soft Computing, Wiley India, 2007.ISBN: 10: 81-265-1075-7.
8. S. Rajasekaran and G.A.V.Pai, Neural Networks, Fuzzy Logic and Genetic Algorithms,PHI, 2003.
9. J.S.R.Jang, C.T.Sun and E.Mizutani, Neuro-Fuzzy and Soft Computing, PHI, 2004,Pearson Education.

Third Year B.C.A. (Under Science) Semester VI

Course Code: BCA601 **Course Title: Android Programming**

Total Contact Hours: 48 hrs. **Total Credits: 04** **Total Marks: 100**

(60 Lectures)

Teaching Scheme: Theory- 05 Lect./ Week

Course Objectives:

The objective of this course is to understand the Android Operating System and develop applications using Google's Android open-source platform.

UNIT NO.	DESCRIPTION	No. of LECTURES
UNIT 1	**1. Introduction to Android** 1.1. Overview 1.2. History 1.3. Features of Android 1.4. Architecture of Android • Overview of Stack • Linux Kernel • Native Libraries • Android Runtime • Application Framework • Applications 1.5. SDK Overview • Platforms • Tools – (JDK, SDK, Eclipse/Android Studio, ADT, AVD, Android Emulator) • Versions 1.6. Creating your first Android Application	**06**
UNIT 2	**2. Activities, Fragments and Intents** 2.1. Introduction to Activities 2.2. Activity Lifecycle 2.3. Introduction to Intents 2.4. Linking Activities using Intents 2.5. Calling built-in applications using Intents 2.6. Introduction to Fragments 2.7. Adding Fragments Dynamically 2.8. Lifecycle of Fragment 2.9. Interaction between Fragments	**09**
UNIT 3	**3. Android User Interface** 3.1. Understanding the components of a screen • Views and ViewGroups • LinearLayout • AbsoluteLayout	**10**

	• TableLayout • RelativeLayout • FrameLayout • ScrollLayout • ScrollView 3.2. Adapting to Display Orientation • Anchoring Views • Resizing and Repositioning 3.3. Managing Changes to Screen Orientation • Persisting State Information during Changes in Configuration • Detecting Orientation Changes • Controlling the Orientation of the Activity 3.4. Utilizing Action Bar • Adding Action Items to the Action Bar • Customizing the Action Items and Application Icon	
UNIT 4	**4. Designing Your User Interface with Views** 4.1. Using Basic Views • TextView • Button, ImageButton, EditText, CheckBox • ToggleButton, RadioButton, and RadioGroup Views • ProgressBar View • AutoCompleteTextView View 4.2. Using Picker Views • TimePicker View • DatePicker View 4.3. Using List Views to Display Long Lists • ListView View • Using the Spinner View 4.4. Understanding Specialized Fragments • Using a ListFragment • Using a DialogFragment • Using a PreferenceFragment	**10**
UNIT 5	**5. Displaying Pictures and Menus** 5.1. Using Image Views to Display Pictures • Gallery and ImageView views • Image Switcher • Grid View 5.2. Using Menus with Views • Creating the helper methods • Options Menu • Context Menu	**05**
UNIT 6	**6. Databases – SQLite** 6.1. Introduction to SQLite 6.2. SQLiteOpenHelper and SQLiteDatabase 6.3. Creating , opening and closing database 6.4. Working with cursors, Insert, Update, Delete 6.5. Building and executing queries	**06**

UNIT 7	**7. Messaging and E-mail** 7.1. SMS Messaging • Sending SMS Messages Programmatically • Getting Feedback after Sending a Message • Sending SMS Messages Using Intent • Receiving SMS Messages • Caveats and Warnings 7.2. Sending E-mail	**06**
UNIT 8	**8. Location-Based Services and Google Map** 8.1. Display Google Maps • Creating the project • Obtaining the Maps API Key • Displaying the Map • Displaying the Zoom Control • Changing Views • Navigating to a specific location • Adding Markers • Getting the location that was touched • Geocoding and Reverse Geocoding 8.2. Getting Location Data 8.3. Monitoring a Location	**08**

Reference Books:

1. Beginning Android4 Application Development, By Wei-Meng Lee WILEY India Edition WROX Publication
2. Professional Android 4 Application Development, By Reto Meier WROX Publication
3. The official site for ***Android developers* -** https://developer.android.com

Third Year B.C.A. (Under Science) Semester VI

Course Code: BCA602 **Course Title: Python Programming**

Total Contact Hours: 48 hrs. (60 Lectures) **Total Credits: 04** **Total Marks: 100**

Teaching Scheme: Theory- 05 Lect./ Week

Course Objectives:

- To introduce various concepts of programming to the students using Python.
- Students should be able to apply the problem solving skills using Python

Unit No.	Contents	No. of Lectures
Unit 1	**Introduction to Python Scripting** • Why Scripting is Useful in Computational Science • Classification of Programming Languages • Productive Pairs of Programming Languages • Gluing Existing Applications • Scripting Yields Shorter Code, Efficiency • Type-Specification (Declaration) of Variables • Flexible Function Interfaces • Interactive Computing • Creating Code at Run Time • Nested Heterogeneous Data Structures • GUI Programming • Mixed Language Programming • When to Choose a Dynamically Typed Language • Why Python? Script or Program? • Application of Python • Concept (immutable)	**04**
Unit 2	**Basic Python** • Python identifiers and reserved words • Lines and indentation, multi-line statements • Comments • Input/output with print and input functions, • Command line arguments and processing command line arguments • Standard data types - basic, none, Boolean (true & False), numbers • Python strings • Data type conversion • Python basic operators (Arithmetic, comparison, assignment, bitwise logical) • Python membership operators (in & not in) • Python identity operators (is & is not) • Operator precedence • Control Statements, Python loops, Iterating by	**06**

	subsequence index, loop control statements (break, continue, pass) • Mathematical functions and constants (import math), Random number functions	
Unit 3	**Python strings** • Concept, escape characters • String special operations • String formatting operator • Single quotes, Double quotes, Triple quotes • Raw String, Unicode strings, Built-in String methods. • Python Lists - concept, creating and accessing elements, updating & deleting lists, basic list operations, reverse • Indexing, slicing and Matrices • built-in List functions • Functional programming tools - filter(), map(), and reduce() • Using Lists as stacks and Queues, List comprehensions	**06**
Unit 4	**Python tuples and sets** • Creating & deleting tuples • Accessing values in a tuple • Updating tuples, delete tuple elements • Basic tuple operations • Indexing, slicing and Matrices, built- in tuple functions. • Sets - Concept, operations.	**06**
Unit 5	**Python Dictionary** • Concept (mutable) • Creating and accessing values in a dictionary • Updating dictionary, delete dictionary elements • Properties of dictionary keys • built-in dictionary functions and methods.	**04**
Unit 6	**Functions** • Defining a function (def) • Calling a function • Function arguments - Pass by value, Keyword Arguments, default arguments • Scope of variable - basic rules • Documentation Strings • Variable Number of Arguments • Call by Reference • Order of arguments (positional, extra & keyword) • Anonymous functions • Recursion • Treatment of Input and Output Arguments • Unpacking argument lists • Lambda forms • Function Objects • function ducktyping & polymorphism	**08**

	• Generators (functions and expressions) and iterators, list comprehensions	
Unit 7	**Files and Directories** • Creating files • Operations on files (open, close, read, write) • File object attributes, file positions, Listing Files in a Directory • Testing File Types • Removing Files and Directories • Copying and Renaming Files • Splitting Pathnames • Creating and Moving to Directories • Traversing Directory Trees • Illustrative programs: word count, copy file	**06**
Unit 8	**Python Classes / Objects** • Object oriented programming and classes in Python - creating classes, instance objects, accessing members • Data hiding (the double underscore prefix) • Built-in class attributes • Garbage collection : the constructor • Overloading methods and operators • Inheritance - implementing a subclass, overriding methods • Recursive calls to methods • Class variables, class methods, and static methods	**08**
Unit 9	**Python Exceptions** • Exception handling : assert statement • Except clause - with no exceptions and multiple exceptions • Try - finally, raising exceptions, user-defined exceptions.	**02**

Reference Books:

1. Introducing Python- Modern Computing in Simple Packages – Bill Lubanovic, O‘Reilly Publication
2. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress
3. Practical Programming: An Introduction to Computer Science Using Python 3, Paul Gries, et al., Pragmatic Bookshelf, 2/E 2014
4. Introduction to Computer Science Using Python- Charles Dierbach, Wiley Publication Learning with Python “, Green Tea Press, 2002
5. E-Books : python_tutorial. pdf, python_book_01.pdf
6. Beginning Programming with Python for Dummies Paperback – 2015 by John Paul Mueller
7. A Beginner’s Python Tutorial: http://en.wikibooks.org/wiki/A Beginner%27s Python Tutorial.

Third Year B.C.A. (Under Science) Semester VI

Course Code: BCA603 **Course Title: Recent Trends in IT (Internet of Things)**

Total Contact Hours: 48 hrs. (60 Lectures) **Total Credits: 04** **Total Marks: 100**

Teaching Scheme: Theory- 05 Lect./ Week

Pre-Requisite: Basic understanding of electronics and microprocessors.
Course Objectives:

1. The Internet of Things (IoT) is aimed at enabling the interconnection and integration of the physical world and the cyber space.
2. To learn about SoC architectures, programming Raspberry Pi and implementation of internet of things and protocols.

Expected Learning Outcomes:
1. Enable learners to understand System On Chip Architectures.
2. Introduction and preparing Raspberry Pi with hardware and installation.
3. Learn physical interfaces and electronics of Raspberry Pi and program them using practical's
4. Learn how to design IoT based prototypes.

Unit No.	Contents	No. of Lecctures
Unit 1	**System on Chip (SoC) and Internet of Things (IoT) Overview** - **System on Chip:** What is System on chip? Structure of System on Chip. - **SoC products:** Field Programmable Gate Array **(**FPGA)**,** General Purpose Graphics Processing Units **(**GPU), Accelerated Processing Unit (APU), Compute Units. -**The IoT paradigm** giving overview of IoT supported Hardware platforms such as: Raspberry pi, SoC on ARM 8 Processors, Arduino and Intel Galileo boards. -**Network Fundamentals**: Wired Networking(Router, Switches), Wireless Networking(Access Points) -**Introduction to Raspberry Pi:** Introduction to Raspberry Pi, Raspberry Pi Hardware, Preparing your raspberry Pi. -**Raspberry Pi Boot:** Learn how this small SoC boots without BIOS. Configuring boot sequences and hardware. -**Introduction to IoT:** What is IoT? IoT examples, Simple IoT LED Program. -**IoT and Protocols** -**IoT Security:** HTTP, UPnp, CoAP, MQTT, XMPP**.** -**IoT Service as a Platform:** Clayster, Thinger.io, SenseIoT, carriots and Node RED. -**IoT Security and Interoperability:** Risks, Modes of Attacks, Tools for Security and Interoperability.	**20**

Unit 2	**Programming Raspberry Pi** **Raspberry Pi and Linux:** About Raspbian, Linux Commands, Configuring Raspberry Pi with Linux Commands **Programing interfaces:** Introduction to Node.js, Python. **Raspberry Pi Interfaces:** UART, GPIO, I2C, SPI **Useful Implementations:** Cross Compilation, Pulse Width Modulation, SPI for Camera.	**15**
Unit 3	**Case Study & advanced IoT Applications:** IoT applications in home, infrastructures, buildings, security, Industries, Home appliances, other IoT electronic equipments. Use of Big Data and Visualization in IoT, Industry 4.0 concepts. Sensors and sensor Node and interfacing using any Embedded target boards (Raspberry Pi / Intel Galileo/ARM Cortex/ Arduino)	**15**
Unit 4	**Internet of Things Privacy, Security and Governance** Introduction, Overview of Governance, Privacy and Security Issues, Contribution from FP7 Projects, Security, Privacy and Trust in IoT-Data-Platforms for Smart Cities, First Steps Towards a Secure Platform, Smartie Approach. Data Aggregation for the IoT in Smart Cities, Security	**10**

TEXT BOOKS:

1. 6LoWPAN: The Wireless Embedded Internet, Zach Shelby, Carsten Bormann, Wiley
2. Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, Dr. Ovidiu Vermesan, Dr. Peter Friess, River Publishers
3. Interconnecting Smart Objects with IP: The Next Internet, Jean-Philippe Vasseur, Adam Dunkels, Morgan Kuffmann
4. Internet of Things : A hands- on Approach by Arshdeep Bahga, Vijay Madisetti
5. IoT Programming: A Simple and Fast Way of Learning IOT by David Etter

REFERENCES:

1. The Internet of Things: From RFID to the Next-Generation Pervasive Networked Lu Yan, Yan Zhang, Laurence T. Yang, Huansheng Ning
2. Internet of Things (A Hands-on-Approach) , Vijay Madisetti , Arshdeep Bahga
3. Designing the Internet of Things , Adrian McEwen (Author), Hakim Cassimally
4. “Mobile Computing,” Tata McGraw Hill, Asoke K Talukder and Roopa R Yavagal, 2010.
5. Computer Networks; By: Tanenbaum, Andrew S; Pearson Education Pte. Ltd., Delhi, 4th Edition
6. Data and Computer Communications; By: Stallings, William; Pearson Education Pte. Ltd., Delhi, 6th Edition
7. “Fundamentals of Mobile and Pervasive Computing,” F. Adelstein and S.K.S. Gupta, McGraw Hill, 2009.
8. Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010

Third Year B.C.A. (Under Science) Semester V or VI

Course Code: BCA-604 **Course Title: Data Analytics**

Total Contact Hours: 48 hrs. **Total Credits: 04** **Total Marks: 100**
(60 Lectures)

Teaching Scheme: Theory- 05 Lect./ Week

Course Objectives:

1. Able to apply fundamental algorithmic ideas to process data.
2. Learn to apply hypotheses and data into actionable Predictions.

Unit No.	Contents	No. of Lectures
Unit 1	• Introduction to data Science o Basics of Data o What is Data Science? o Data science process o Stages in data science project • Basics of Data Analytics • Types of Analytics – Descriptive, Predictive, Prescriptive • Statistical Inference o Populations and samples o Statistical modeling, o Probability o Distribution o Correlation o Regression	10
Unit 2	Introduction to Machine Learning • Basics of Machine Leaning • Supervised Machine Learning ▪ K- Nearest-Neighbors, ▪ Naïve Bayes ▪ Decision tree ▪ Support Vector Machines • Unsupervised Machine Learning ▪ Cluster analysis ▪ K means ▪ Association Rule Mining • Apriori algorithms • Regression Analysis ▪ Linear Regression ▪ Nonlinear Regression	25
Unit 3	Data Analytics with Python Programming □ Numpy o Arrays o Array indexing o Datatypes o Array math o Broadcasting	15

	SciPy o Image operations o Distance between points Data analysis and manipulation using Pandas package o Importing Data , Creating A DataFrame, o DataFrame Methods, o Indexing DataFrames, Boolean Indexing o Indexing Using Labels , Multi-Indexing o Merge DataFrames o Sorting DataFrames o Apply Function o Pivot Table, Crosstab o Iterating over rows of a dataframe	
Unit 4	Data Visualization ▪ Basic principles, ▪ Ideas and tools for data visualization ▪ Graph Visualization, ▪ Data Summaries, ▪ Model Checking & Comparison ▪ Purpose of visualization ▪ Multidimensional visualization ▪ Tree visualization ▪ Graph visualization ▪ Visualization techniques ▪ Understanding analytics output and their usage ▪ Scikit package ▪ matplotlib library o Plotting o Subplots o Images	10

Reference Books:

1. The elements of statistical learning. Hastie, Trevor, et al., Vol. 2. No. 1. New York: springer, 2009.
2. Applied statistics and probability for engineers. Montgomery, Douglas C., and George C. Runger. John Wiley & Sons,2010
3. Scaling up Machine Learning to White “Hadoop: The Definitive Guide” Third Edition, Bekkerman et al., O‟reilly Media, 2012.
4. “Mining of Massive Datasets”, Anand Rajaraman and Jeffrey David Ullman, Cambridge University Press, 2012.
5. Developing Analytic Talent: Becoming a Data Scientist, Vincent Granville, wiley, 2014.
6. Introduction to Data Science, Jeffrey Stanton & Robert De Graaf, Version 2.0, 2013.
7. “Practical Data Science with R”, Nina Zumel, John Mount, Manning Publications, 2014.
8. “Mining of Massive Datasets”, Jure Leskovec, Anand Rajaraman, Jeffrey D.Ullman, Cambridge

University Press, 2014.

9. “Beginning R - The Statistical Programming Language”, Mark Gardener, John Wiley & Sons, Inc., 2012.
10. “An Introduction to R”, W. N. Venables, D. M. Smith and the R Core Team, 2013.
11. “Practical Data Science Cookbook”, Tony Ojeda, Sean Patrick Murphy, Benjamin Beng fort, Abhijit Dasgupta, Packt Publishing Ltd., 2014.
12. “Visualize This: The Flowing Data Guide to Design, Visualization, and Statistics”, Nathan Yau, Wiley, 2011.
13. “Professional Hadoop Solutions”, Boris lublinsky, Kevin t. Smith, Alexey Yakubovich, Wiley, ISBN: 9788126551071, 2015.
14. http://www.johndcook.com/R_language_for_programmers.html
15. http://bigdatauniversity.com/
16. http://home.ubalt.edu/ntsbarsh/stat-data/topics.htm#rintroduction

Third Year B.C.A. (Under Science) Semester VI

BCA – 605 Course **Title: Android Programming Lab**
Course I

Total Contact Hours: 48 hrs. **Total Credits: 04**

Total Marks: 100

Note that these are only sample assignments. Teachers may conduct practicals by preparing similar types of examples

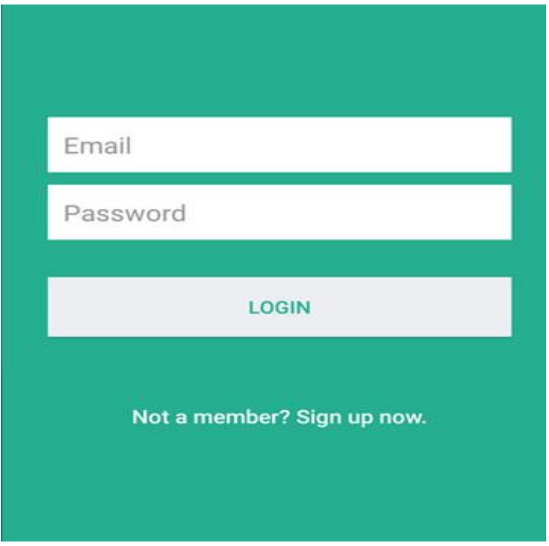
Sample Assignments on Android Programming

Assignment 1: Introduction to Android

1. Install Android Studio and build simple Hello World application.

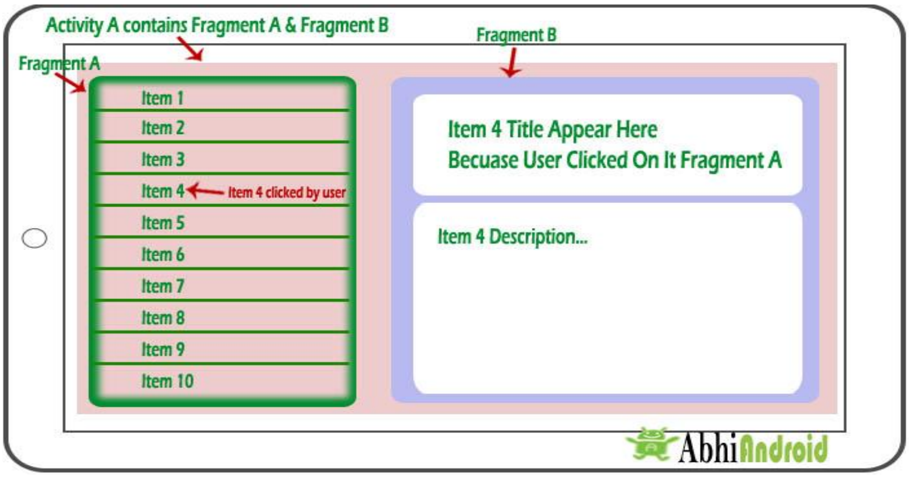
Assignment 2: Activities, Fragments and Intents

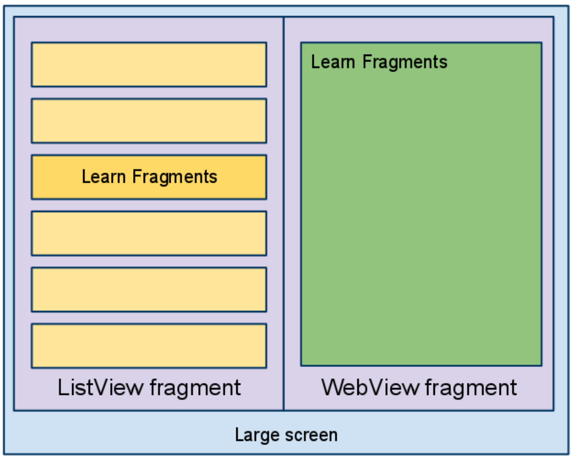
1. Design Login Activity shown below.

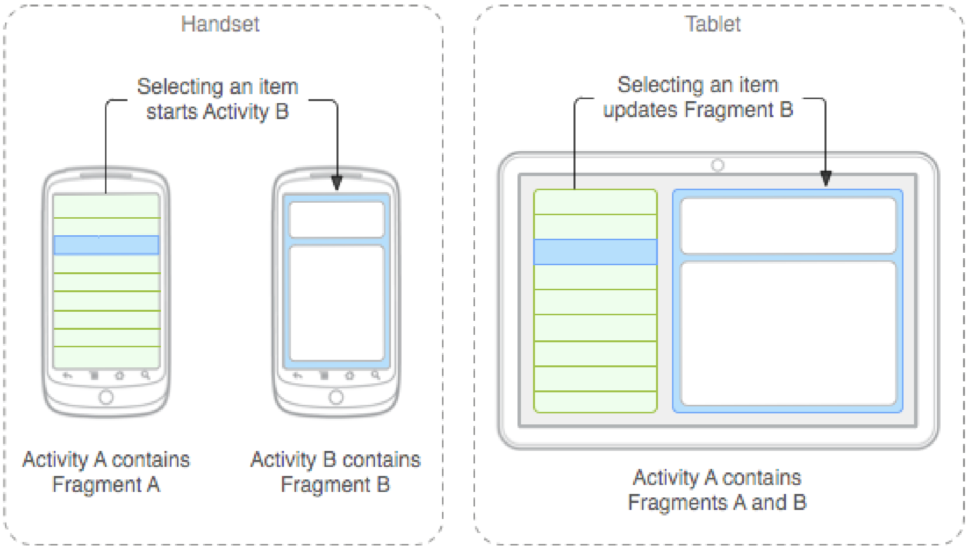


Not a member? Sign up now.

2. Create application to display details of selected list item on second activity (Use Fragmentation).



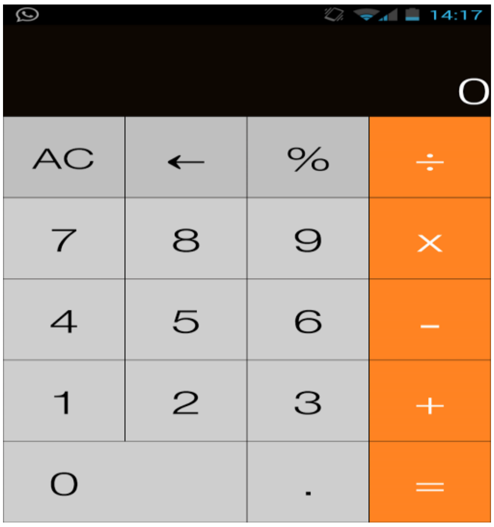



Handset
Selecting an item starts Activity B
Activity A contains Fragment A
Activity B contains Fragment B
Tablet
Selecting an item updates Fragment B
Activity A contains Fragments A and B

3. Create first activity to accept information like first name, last name, date of birth, email-id and display all information on second activity when user click on submit button.

Assignment 3: Android User Interface and Event Handling

1. Create the simple calculator shown below. Also, perform appropriate operations.

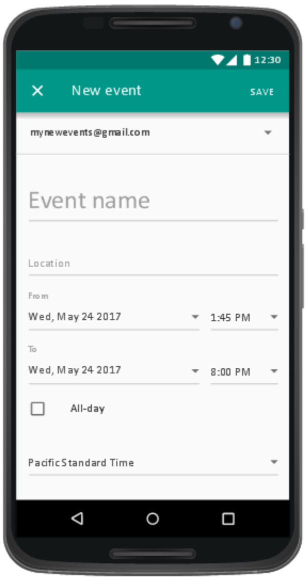


2. Create application to calculate GPA.
3. Create chat application.

Assignment 4: Designing Your User Interface with Views

1. Create a custom "Contact" layout to hold multiple pieces of information, including: Photo, Name, Contact Number, E-mail id.

2. Create application to demonstrate date and time picker.



mynewevents@gmail.com

Event name

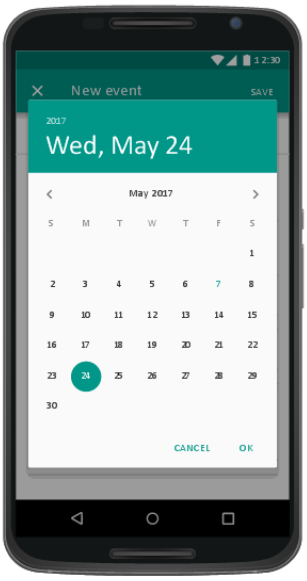
Location

From
Wed, May 24 2017 1:45 PM

To
Wed, May 24 2017 8:00 PM

□ All-day

Pacific Standard Time



Assignment 5: Displaying Pictures and Menus

1. Construct an app that toggles a light bulb on and off when the user clicks on toggle button.
2. Create application as shown below.

3. Create gallery application to display all images date wise (Use Grid View).

Assignment 6: Databases – SQLite

1. Create login activity (referAssignment 2 Example 1). If Email and password matches with database display "login successful" message else display error message.
2. Construct a simple notes list that lets the user add new notes but not edit them. Demonstrates the basics of ListActivity.Use a SQLite database to store the notes.
3. Create tables: Course (id, name, instructor) and Student (id, name).
   Course and Student have a many to many relationship. Create a GUI based system for performing the following operations on the tables:
    1. Course: Add Course, View All students of a specific course
    2. Student: Add Student, Delete Student, View All students, Search student

Assignment 7: Messaging and E-mail

1. Create application to send and receive messages.
2. Create application to send email with validation.
3. Create application to send email with attachment.

Assignment 8: Location-Based Services and Google Map

1. Write a program to find the current location of an Android device and display details of the place like Street name, city with Geocoding.
2. Write a program to track android device usingGoogle Maps.
3. Write a program todraw path along a route in Google map.

Third Year B.C.A. (Under Science) Semester VI

Course BCA606 **Title: Python Lab Cource II**

Total Contact Hours: 48 hrs. **Total Credits: 04**

Total Marks: 100

Python Assignments:

1. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old.
2. Write a program to check whether the number is even or odd, print out an appropriate message to the user.
3. Write a program which will find all such numbers which are divisible by 7.
4. Write a program which can compute the factorial of a given numbers.
5. Write a program that prints out all the elements of the list that are less than 10.
6. Write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes.
7. Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number.Given the input: 34,67,55,33,12,98

   Then, the output should be:

   ```
   ['34', '67', '55', '33', '12', '98']
   ('34', '67', '55', '33', '12', '98')
   ```

8. Make a two-player Rock-Paper-Scissors game. (*Hint: Ask for player plays (using input), compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game*) Rules:
   Rock beats scissors
   Scissors beats paper
   Paper beats rock
9. To determine whether the number is prime or not.
10. To check whether a number is palindrome or not. (using recursion and without recursion).
11. Write a program (function!) that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates.Write two different functions to do this - one using a loop and constructing a list, and another using sets.
12. Write a program that asks the user how many Fibonnaci numbers to generate and then generates them.
13. Write a program (using functions!) that asks the user for a long string containing multiple words. Print back to the user the same string, except with the words in backwards order. E.g “ I am tybca student” is :”student tybca am I”
14. Write a password generator in Python. Be creative with how you generate passwords - strong passwords have a mix of lowercase letters, uppercase letters, numbers, and symbols. The passwords should be random, generating a new password every time the user asks for a new password.
15. Write a program to implement binary search to search the given element using function.
16. Given a `.txt` file that has a list of a bunch of names, count how many of each name there are in the file, and print out the results to the screen.

17. Implement a function that takes as input three variables, and returns the largest of the three.(do not use max function)

18. Create a dictionary (in your file) of names and birthdays. When you run your program it should ask the user to enter a name, and return the birthday of that person back to them.

19. Write a program that takes a list of numbers (for example, `a = [5, 10, 15, 20, 25]`) and makes a new list of only the first and last elements of the given list.

20. Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized.

21. Write a program that accepts a sentence and calculate the number of letters and digits.

22. Write a program that accepts a sentence and calculate the number of upper case letters and lower case letters.

String:

A string is a sequence of characters. The string is a sequence of Unicode character in Python. Unicode was introduced to include every character in all languages and bring uniformity in encoding.

Strings can be created by enclosing characters inside a single quote or double quotes. Even triple quotes can be used in Python but generally used to represent multiline strings and docstrings.

```
# All of the following are equivalent
my_string = 'Hello'
print(my_string)
my_string = "Hello"
print(my_string)
my_string = '''Hello'''
print(my_string)
# triple quotes string can extend multiple lines
my_string = """Hello, welcome to
          the world of Python"""
print(my_string)
```

The output of *stringm.py* will be:

```
Hello
Hello
Hello
Hello, welcome to
          the world of Python
```

To access characters in a string:

We can access individual characters using indexing and a range of characters using slicing. Index starts from 0. Trying to access a character out of index range will raise an IndexError. The index must be an integer. We can't use float or other types, this will result into TypeError. Python allows negative indexing for its sequences.The index of -1 refers to the last item, -2 to the second last item and so on. We can access a range of items in a string by using the slicing operator (colon).

```
str = 'programing'
print('str = ', str)
```

```
#first character
print('str[0] = ', str[0])
#last character
print('str[-1] = ', str[-1])
#slicing 2nd to 5th character
print('str[1:5] = ', str[1:5])
#slicing 6th to 2nd last character
print('str[5:-2] = ', str[5:-2])
```

Update string:

The existing string can be update by (re)assigning a variable to another string. The new value can be related to its previous value or to a completely different string altogether. For example −

```
var1 = 'Hello World!'
print "Updated String :- ", var1[:6] + 'Python'
```

output:

```
Updated String :-  Hello Python
```

Python includes the following built-in methods to manipulate strings −

Sr.No.	Methods with Description
1	**capitalize()** Capitalizes first letter of string
2	**center(width, fillchar)** Returns a space-padded string with the original string centered to a total of width columns.
3	**count(str, beg= 0,end=len(string))** Counts how many times str occurs in string or in a substring of string if starting index beg and ending index end are given.
4	**decode(encoding='UTF-8',errors='strict')** Decodes the string using the codec registered for encoding. encoding defaults to the default string encoding.
5	**encode(encoding='UTF-8',errors='strict')** Returns encoded string version of string; on error, default is to raise a ValueError unless errors is given with 'ignore' or 'replace'.
6	**endswith(suffix, beg=0, end=len(string))** Determines if string or a substring of string (if starting index beg and ending index end are given) ends with suffix; returns true if so and false otherwise.
7	**expandtabs(tabsize=8)** Expands tabs in string to multiple spaces; defaults to 8 spaces per tab if tabsize not provided.
8	**find(str, beg=0 end=len(string))** Determine if str occurs in string or in a substring of string if starting index beg

	and ending index end are given returns index if found and -1 otherwise.
9	**index(str, beg=0, end=len(string))** Same as find(), but raises an exception if str not found.
10	**isalnum()** Returns true if string has at least 1 character and all characters are alphanumeric and false otherwise.
11	**isalpha()** Returns true if string has at least 1 character and all characters are alphabetic and false otherwise.
12	**isdigit()** Returns true if string contains only digits and false otherwise.
13	**islower()** Returns true if string has at least 1 cased character and all cased characters are in lowercase and false otherwise.
14	**isnumeric()** Returns true if a unicode string contains only numeric characters and false otherwise.
15	**isspace()** Returns true if string contains only whitespace characters and false otherwise.
16	**istitle()** Returns true if string is properly "titlecased" and false otherwise.
17	**isupper()** Returns true if string has at least one cased character and all cased characters are in uppercase and false otherwise.
18	**join(seq)** Merges (concatenates) the string representations of elements in sequence seq into a string, with separator string.
19	**len(string)** Returns the length of the string
20	**ljust(width[, fillchar])** Returns a space-padded string with the original string left-justified to a total of width columns.
21	**lower()** Converts all uppercase letters in string to lowercase.
22	**lstrip()** Removes all leading whitespace in string.

23	**maketrans()** Returns a translation table to be used in translate function.
24	**max(str)** Returns the max alphabetical character from the string str.
25	**min(str)** Returns the min alphabetical character from the string str.
26	**replace(old, new [, max])** Replaces all occurrences of old in string with new or at most max occurrences if max given.
27	**rfind(str, beg=0,end=len(string))** Same as find(), but search backwards in string.
28	**rindex(str, beg=0, end=len(string))** Same as index(), but search backwards in string.
29	**rjust(width,[, fillchar])** Returns a space-padded string with the original string right-justified to a total of width columns.
30	**rstrip()** Removes all trailing whitespace of string.
31	**split(str="", num=string.count(str))** Splits string according to delimiter str (space if not provided) and returns list of substrings; split into at most num substrings if given.
32	**splitlines(num=string.count('\n'))** Splits string at all (or num) NEWLINEs and returns a list of each line with NEWLINEs removed.
33	**startswith(str, beg=0,end=len(string))** Determines if string or a substring of string (if starting index beg and ending index end are given) starts with substring str; returns true if so and false otherwise.
34	**strip([chars])** Performs both lstrip() and rstrip() on string.
35	**swapcase()** Inverts case for all letters in string.
36	**title()** Returns "titlecased" version of string, that is, all words begin with uppercase and the rest are lowercase.
37	**translate(table, deletechars="")**

	Translates string according to translation table str(256 chars), removing those in the del string.
38	**upper()** Converts lowercase letters in string to uppercase.
39	**zfill (width)** Returns original string leftpadded with zeros to a total of width characters; intended for numbers, zfill() retains any sign given (less one zero).
40	**isdecimal()** Returns true if a unicode string contains only decimal characters and false otherwise.

1. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument.
2. Write a Python program to converting an Integer to a string in any base.
3. Write a Python program of recursion list sum.
4. Write a Python program to solve the Fibonacci sequence using recursion.
5. Write a Python program to get the sum of a non-negative integer.
6. Write a Python program to calculate the value of 'a' to the power 'b'
7. Write a Python program to find the greatest common divisor (gcd) of two integers
8. Write a Python function that takes a list and returns a new list with unique elements of the first list.
9. Write a Python function to check whether a number is perfect or not
10. Write a Python program to read a file line by line store it into an array.
11. Write a Python program to count the number of lines in a text file.
12. Write a Python program to count the frequency of words in a file.
13. Write a Python program to copy the contents of a file to another file
14. Write a Python program to read a random line from a file
15. Write a Python class to implement pow(x, n).
16. Write a Python class to reverse a string word by word.
    Input string : 'hello .py'
    Expected Output : '.py hello'
17. Write a Python class named Rectangle constructed by a length and width and a method which will compute the area and perimeter of a rectangle. –
18. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle

Third Year B.C.A. (Under Science) Semester VI

Course Code: BCA607 **Course Title: Introduction to Green Computing**

Total Contact Hours: 24 hrs. (30ectures) **Total Credits: 02** **Total Marks: 50**

Teaching Scheme: Theory- 05 Lect./ Week

Course Objectives:

1. Building more energy-efficient computing systems as well as building computing technology that increases energy-efficiency of other physical systems.
2. Investigate recent advances in the broad realm of green technologies to save energy and reduce the carbon footprint of modern computing and engineered systems.
3. A holistic coverage is given ranging from single device issues to algorithms for reducing power consumption of data centers, transportation systems, and smart buildings.

Unit No.	Contents	No. of Lecctures
Unit 1	**1. Introduction to Green Computing** Websites &statistics How bad the energy crisis really is?	**04**
Unit 2	**2. Reducing the IT footprint** What really contributes to the footprint (from machine manufacturing to disposal)? Saving energy on a single machine Saving energy in networking and other components Saving energy in clusters and data centers Saving energy on data center cooling	**10**
Unit 3	**3. Computing technology for energy efficiency of other physical systems** Computing technology for greener transportation Computing technology for smarter buildings Carbon footprint calculators: what is my footprint?	**10**
Unit 4	**4. Major green initiatives** Sustainable IT, Green Business, Smarter Plant.	**06**

Reference Books:

1. Green Computing: Tools and Techniques for Saving Energy, Money, and Resources1st Edition by Bud E. Smith(CRC Press)
2. The Green Computing Book by Wu-Chun Feng (CRC Press)
3. Green it for sustainable business practice: An ISEB Foundation Guide by Mark G. O’Neill.