

EVALUATION GUIDELINES

PG-DMC August 2019

1. Evaluation

Evaluation is a necessary and essential part of conducting the PG-Diploma in Mobile Computing, as it provides important feedback and inputs to both the institute as well as the student. The institute gets an idea about the relative performance of each student, which also serves as feedback about the design and conduct of the course. The student gets a clear picture of his academic standing, individually and in comparison to his fellow students.

In order to ensure timely and efficient evaluation and certification of all students, the following guidelines are being issued and should be followed religiously.

2. Evaluation Methodology

- 2.1 Each centre should have a Designated Responsible Member (DRM) for Evaluation.
- 2.2 The DRM Evaluation would be responsible for coordinating all activities relating to evaluation at the training centre and for communicating with CDAC ACTS Pune.
- 2.3 Evaluation is a compulsory part of the process of obtaining the PG-DMC. All students are required to pass each module of the course in order to be eligible to receive the PG-Diploma Certificate.
- 2.4 The faculty of every module should outline the objectives of the evaluation to be conducted for that module, so as to enable the student to prepare himself or herself properly.

3. Modular Evaluation

- 3.1 Credit System: The credit-based training and grading system essentially implies a redefining of the curriculum into smaller measurable entities or '**modules**' with the hours required for '**learning**' (not teaching) these being at the primary focus and the development of a mechanism whereby these modules can be combined in different ways so as to qualify for a certificate Diploma. In a sense, therefore, the completion of a single 'module' of learning can pave the way for learning other modules and a combination of modules in keeping with the needs and interests of the learners illustrates the much talked about 'cafeteria approach' to learning with the Learner at the centre state of all academic transactions. Following are the features of Credit System.

- 3.1.1 Learner-centric approach
- 3.1.2 Workload estimated on time invested on learning rather than time invested on teaching
- 3.1.3 Workload=Theory classes + Practical Classes + Assignments + Seminar + Individual Learning + Preparation for examination
- 3.1.4 A program is divided into various modules or units and credit is assigned to each module/unit based on its learn ability.
- 3.1.5 Number of credit points assigned to a module depends upon the number of hours required to learn that module.

- 3.1.6 Credit Transfer Mechanism: Horizontal credit transfer (between similar level programs like Diploma to Diploma), Vertical Credit Transfer (credit transfer to higher level programs like certificate to diploma)
- 3.1.7 Inter institution credit transfer and intra institution credit transfer
- 3.1.8 Assumption: 1 credit = 25 hours of learner's load
- 3.1.9 Credit based evaluation will have following marking system:
 - 40% end term examination + 40% Practical assessment + 20% internal assessment.
 - Between 1-2 credit = 50 marks evaluation,
 - Between 3-6 credit = 100 marks, if there is no lab then 50 marks evaluation (T-40+IA-10)
 - Between 7-10 credit = 150 marks

3.2 Grading system:

The academic performance of a student shall be graded on a ten-point scale. The letter Grades awarded to a student in all the courses shall be converted into Overall Grade Point Average (OGPA) respectively.

The method of evaluation of students in the course and the actual grading of a student's performance are based on the following guidelines.

3.3 Subject wise Evaluation

- 3.3.1 A separate evaluation process is to be conducted for every module of the course.
- 3.3.2 The evaluation for each module must be completed as per guidelines given below. The mid-module /surprise test evaluation is mandatory and can be taken after discussion with the concerned faculty.
- 3.3.3 Students are evaluated on a continuous and throughout the duration of the course to make a fair assessment of the skills acquired by them. To have a very uniform and fair assessment. The evaluation process is divided into two parts:
 - Continuous Assessment - CA (60 marks)
 - Centralized Course Mid and End Examination - CMCE & CCEE (40 Marks)

Continuous Assessment: This is being done primarily by the respective faculty in the form of Lab tests, assignments, quizzes etc conducted (with the help of the respective course coordinators) at regular intervals and as and when the portions of the modules are completed. These are basically internal exams and local to the centre. This process is further categorized into two parts.

 - Lab test (40 marks)
 - Internal test (20 mark): Assignment/Case Studies /quiz and other valuation methods like case study, viva, group discussion depending on the subject and the faculty (20 marks)
- 3.3.4 It is recommended to conduct the Effective Communications & Aptitude sessions for the benefit of the students and also conduct some surprise test for Effective Communications & Aptitude sessions.

- 3.3.5 The figures shown below indicate the weightage of each module in the final performance statement. The examination(s) for each module must be conducted for at least that number of marks. However, the centre may conduct evaluation for a higher number of marks, in which case the marks will be scaled down. For example, if the examination for the Operating Systems Concepts module is conducted for 100 marks, the marks earned by the student will be scaled down to out of 40.
- 3.3.6 The weight age for each component will normally be:
- Theory 40% (Through Centralized Course Mid and End Examination i.e. CMCE and CCEE)
- Laboratory 40%
- Internal 20%
(Internal Assessment: Through Lab Assignment Evaluation, Surprise Tests, attendance, Viva, Seminars)
- Note: Where a module does not have a practical component, the practical component weight age will be merged with the Internal Assessment component weight age.**
- 3.3.7 A student must score a minimum of 40 percent marks in each component of the evaluation, and also in the aggregate score, in order to successfully clear the module. If a student scores more than 40% on aggregate but has scored less than 40% in one component of the evaluation, he will not be declared as passed.
- 3.3.8 The figures shown below indicate the weightage of each subject in the final performance statement for PG-DMC.

Subject Code	Module Name	Contact Hours			Total Workload	Credits	Marks			
		Theory	Lab	Total			Theory (CCEE)	Lab	IA	Total Marks
Term-I (Centralized Mid-Course Examination)										
DMC01	OOPs with C++ Programming	30	30	60	100	4	40	40	20	100
DMC 02	Algorithm and Data Structures	30	30	60	100	4	40	40	20	100
DMC03	OS Concepts and Linux Programming	20	20	40	75	3	40	40	20	100
	Introduction to DBMS	20	20	40	75	3				
DMC04	Java Programming-I(Core Java)	34	36	70	100	4	40	40	20	100
DMC05	Java Programming-II(Advanced Java)	40	40	80	100	4	40	40	20	100
Term-II (Centralized Course End Examination)										
DMC06	iOS Programming	58	56	114	150	6	40	40	20	100
DMC07	Android Programming	68	68	136	150	6	40	40	20	100
DMC08	Hybrid Mobile Apps Programming	40	40	80	150	6	40	40	20	100
DMC09	Effective Communication	50	-	50	75	3	-	-	50	Grade
DMC10	Aptitude and General English	50	-	50	75	3	-	40	10	Grade
DMC11	Project	-	120	120	300	12	-	-	200	Grade

	Total			900	1500	60	320	360	420	800
--	-------	--	--	-----	------	----	-----	-----	-----	-----

3.3.9 Overall 75% attendance is required for a student to become eligible for the CMCE and CCEE.

3.4 General guidelines for award of grades:

The marks of all the components of a subject shall be added to get total marks out of 100. The rounding off shall be done on the higher side.

The grades shall be awarded on the basis of absolute marks, as mentioned in Table-1.

Marks	Grade Points	Grade
90 and above	10	A
80 to Less than 90	9	B
65 to Less than 80	8	C
50 to less than 65	7	D
40 to less than 50	6	E
Less than 40	0	F

3.5 Calculation of Overall Grade Point Average:

OGPA will be calculated by following formula:

$$OGP = \frac{\sum_{i=1}^n C_i G_i}{\sum_{i=1}^n C_i}$$

Where C_i is number of credit hours assigned to a module i and grade point secured by the students is (G_i) in that module.

OGPA is calculated by multiplying the number of credit hours assigned to a module by the value of the grade earned in that module.

Formula of conversion of equivalent Percentage of OGPA:

Percentage marks = (OGPA-0.5)*10

Illustrations of Calculation: - The illustration for the conversion of marks into grades in theory & practical, if any in individual module are as shown below as :-

Subject Code	Module Name	Secured Marks	Total Marks	Credits (C)	Earned Grade	Earned Grade point (G)	C *G	OGPA= $\frac{\sum C *G}{\sum C}$
DAC01	Operating System Concepts	90	100	4	A	10	40	524/60=8.33
DAC02	OOPs with C++ Programming	85	100	4	B	9	36	
DAC03	Algorithm & Data structures	70	100	4	C	8	32	
DAC04	Software Application Development Tools &	40	50	3	B	9	27	

	Techniques							
DAC05	Advanced Web Programming	75	100	3	C	8	24	
DAC06	JavaScript Framework	70	100	4	C	8	32	
DAC07	Database Technologies	80	100	3	B	9	27	
DAC08	Java Technologies-I (Core Java)	75	100	5	C	8	40	
DAC09	Java Technologies-II (Web Based Java)	85	100	6	B	9	54	
DAC10	Microsoft .NET	75	100	4	C	8	32	
DAC11	Application Security & Testing	75	100	4	C	8	32	
	Upcoming Technology (Parallel Computing)							
DAC12	Effective Communication	70	50	3	C	8	24	
DAC13	Aptitude & General English	70	50	3	C	8	24	
DAC14	Project	190	200	10	A	10	100	
	Total	1150	1350	$\sum C = 60$			$\sum C * G = 524$	

Then conversion of equivalent Percentage of OGPA = $(OGP - 0.5) * 10 = (8.33 - 0.5) * 10 = 78.3 \%$

4. Centralized Mid-Course and Centralized Course End Examination (CMCE & CCEE):

Centralized Course Exam shall be conducted in two phases i.e. **Centralized Mid-Course Exam (CMCE) & Centralized Course End Exam (CCEE)**, which will test the knowledge of the students about the theoretical concepts of each module and it is a compulsory part of the evaluation. Conducting CCEE involves performing duty with responsibility. A small mistake in the process may hamper the whole system. Everyone has to play their role in an effective manner. It is a joint effort work which has to be carried out in a combined way. Right from receiving the question paper from the national resource centres (NRCs) to sending the answer sheet for evaluation dealt with lot of responsibility.

4.1 Guidelines of CMCE & CCEE:

CMCE and CCEE will be conducted normally before the commencement of Project work of PG-DMC.

The written examination should be of 60 minutes duration. It should consist of objective questions. A typical objective type exam paper should contain the following types of questions: –

- ° Multiple choice
- ° Yes or No
- ° True or False

Objective questions are useful in testing the recognition and recall abilities of students. They also help in keeping the exam short and easier to evaluate.

For the pure objective type question papers, there will be 40 objective type questions with 4 maximum answer options having only one correct option. The value of each objective type question is of one mark only. There will not be any negative marks for the wrong answers given by the students.

4.2 Guidelines for setting Question Papers

While setting the question papers for theory Exam the following weightages should be assigned as per the difficulty level of the questions.

Levels	Requirements	Weightage
Level A - Easy	Requires elementary knowledge which may be obtained by attending all lectures and completion of mandatory lab assignments	25%
Level B - Intermediate	Requires thorough study of all course material, attendance at all lectures and completion of mandatory assignments	50%
Level C - difficult	Requires study and lab work beyond the prescribed course material and mandatory assignments	25%

4.3 Guidelines for generating theory questions

- 4.3.1 Question paper setter has to use sample paper format provided by C-DAC, ACTS Pune
- 4.3.2 Mention the module name without fail.
- 4.3.3 Language of the question should be easy to understand.
- 4.3.4 The answers must have relevant objective type choices and “only one” correct answer.
- 4.3.5 The questions must be prepared by referring appropriate books, reference books, reference material, and course material having good information.
- 4.3.6 The question must be created by the domain expert afresh and should not be copied directly from any book, website, existing previous question papers etc.
- 4.3.7 The question should be unique and should have not been published anywhere.
- 4.3.8 Please mention the source of the question wherever possible, as it may help us in referring the same for detailing if required.
- 4.3.9 The caliber of the question should suffice the growing need of competition.
- 4.3.10 The question paper should have questions covering the entire syllabus.
- 4.3.11 The questions have to be typed in MS Word with “Arial” having letter size 12 point. Do not bold any letter, word or sentence in any part of the question paper.
- 4.3.12 It is essential to give password to the word document and send/tell the password separately.
- 4.3.13 It is essential that utmost care is taken at your end to maintain the secrecy of the soft copy at all time.
- 4.3.14 An expert team will review all questions. The questions will be filtered as per following:
 - °If the question is incomplete
 - °If the answer of the question is wrong
 - °If the question is not there in the syllabus
 - °If the question appears more than once
 - °If the question is too lengthy
 - °If the question is irrelevant
 - °If the options to the questions are irrelevant

4.4 Template for generation of Questions

Date:

Question generated by: Mr. /Ms.

Topic / Module:

Q. No.

Question:

<Text of the question>

Format Type 1 (If answer choices are lengthy, one or more than one line)

Answer Choices

A:

B:

C:

D:

Format Type 2 (If answer choices are short, two to three words)

Answer Choices

A:

B:

C:

D:

Format Type 3 (If answer choices are very short, one word only)

Answer Choices

A:

B:

C:

D:

Difficulty Level: Easy / Intermediate / Difficult

Reference: (Name of books)

(If question taken from book) (Mention name of the book, author, ISBN)

Total Number of Questions Generated: _____

4.5 Template for Answer Key:

Module name			
Question No.	Answer Keys	Question No.	Answer Keys
1		26	
2		27	
3		28	
4		29	
5		30	
6		31	
7		32	
8		33	
9		34	
10		35	
11		36	
12		37	
13		38	
14		39	
15		40	
16		41	
17		42	
18		43	
19		44	
20		45	
21		46	
22		47	
23		48	
24		49	
25		50	

4.6 Evaluation of Lab Exam should be based on the following criteria:

Criteria	Details	Max Marks
Design	Correctness of paper design as per the given problem statement & Problem understanding.	6
	Presentation of the design work	4
Coding	Correctness of Design Coding, Syntax Correctness	5
	Correctness of Test Bench or do file, Syntax correctness	5
Simulation and Implementation	The program produces desired output for set of inputs and handles all valid and Invalid inputs	7
	Syntax of UCF files and other synthesis related constructs	3
Viva/Quiz	Explanation of the code written in practical exam / Quiz based on language constructs	10
	Total	40

5. Evaluation of answer papers:

Centre needs to evaluate the students manually and process the result. All the answers written by the student should be read carefully while evaluating the answer sheets. Since most questions will be technical in nature, a brief scan of the answers will not suffice if there are technical details in the answers.

For objective type multiple choice questions centers can use of OMR or OCR sheets for processing the result. Centre which are not using OMR or OCR can evaluate the students manually and process the result.

If a student requests for re-evaluation then the student has to pay Rs. 150 /- for each module and it should be routed through training centre. The Re-evaluation fee should be paid to the respective C-DAC training Centres, in case of Authorized Training Centres associated to C-DAC, Pune, payment to be made in favour of "C-DAC, ACTS" and payable at Pune. (This is applicable only for theory exam)

6. Moderation:

Grace marks would be awarded as per the methodology below:

- Maximum of 4% of total course end theory exam marks can be awarded to a candidate.
- Maximum of 8% of individual course end module test marks (maximum marks) can be awarded per module.

S No.	Name of the course	Total Marks	Maximum grace marks for the course	Maximum Marks per module	Maximum grace marks per module
1	PG-DMC	320	13	40	3

On completion of the moderation exercise the revised marks should be updated in the marks database.

7. Re-examinations:

The following conditions will be applicable for the course end re-exam:

- 7.1. Students who do not appear for an exam on the scheduled date will not have an automatic right to reexamination. Only those students who, in the opinion of the centre/course coordinator have a genuine reason for being absent may be allowed to appear for a re-exam.
- 7.2. a. Students who do not appear for an exam on the scheduled date will not have an automatic right to reexamination. Only those students who, in the opinion of the centre/course coordinator have a genuine reason for being absent may be allowed to appear for a re-exam.
- 7.3. Students who have failed an exam may be allowed to appear for a re-exam.
- 7.4. The re-exam should be conducted following the same process as the regular examination.
- 7.5. Students, who failed/remained absent in the Course End Examination conducted by C-DAC, shall be allowed to appear in the re-examination only once.
- 7.6. Students who remain absent or fail in the re-examination will not get any further chance for appearing for the re-examination. In such case the candidate can receive the Performance Statement and the certificate of participation without any grade.
- 7.7. On evaluation of their answer sheets 20% of the marks obtained by the students will be deducted (towards de-rating for re-examination) for arriving at the final score, i.e. in order to clear the module test the student has to score a minimum of 48% marks instead of 40%.
- 7.8. The fee for the re-exam is currently NIL.
- 7.9. There will be no re-exam for the re-exam

8. Evaluation Guidelines for Effective Communication

There should be module end exam in Effective Communications (EC) for the benefit of the students. Total marks for evaluation will be 100 for Effective Communication/Presentation / Seminar / GD / Attendance. After evaluation marks need to be converted to grades as per the scale given below, which will be given in the mark sheet.

8.1 Evaluation method:

Sl. No.	Module Name	CCEE Marks	IA Marks	Total Marks
1	Communication skills	0	50	100
2	Presentation / Seminar/GD	0	40	
3	Attendance for EC sessions	0	10	

8.2 Internal evaluation Method for Effective Communication

The examination for this module must be conducted for at least that number of marks. However, the centre may conduct evaluation for a higher number of marks, in which case the marks will be scaled down. For example, if the examination for Presentation is conducted for 100 marks, the marks earned by the student will be scaled down to out of 40.

It is recommended to conduct some surprise tests for Effective Communications & Aptitude sessions as Internal marks.

Centres can evaluate on Test/Assignment/Presentation/Seminar/Group discussion and may depend on subject & faculty.

8.3 Guidelines for Presentation/Seminar/Group Discussion Evaluation

Evaluation of Presentation, Seminar and Group Discussion needs to be carried out as per the following guidelines.

- a. Submission of their seminar topic along with the references.
- b. Two presentations each for 15 minutes on the seminar restricted to 15 – 20 slides followed by evaluation.
- c. Presentation/Seminar evaluation for 100 marks will be split up as follows:
 - i. Communication skills – 20
 - ii. Use of verbal and non verbal language – 10
 - iii. Presentation skills – 20
 - iv. Contents of the seminar/GD – 10
 - v. Contents Flow of Presentation – 10
 - vi. Depth of Knowledge in the topic – 10
 - vii. Viva Voce – 10
 - viii. Attendance – 10

9. Project Module:

For all the projects that the students do during the course, the evaluation will proceed roughly on the lines of team assignments. However, a group of teams can form a larger group to implement more complex designs. The projects have to be evaluated for the following heads:

- 9.1. **Project Document:** Refer to the project document guideline for details of what is required in a project document. The quality of the design also affects the credit in this head.

(15 points)

- 9.2. **Implementation:** Under this head the actual work done has to be evaluated. This has to be done for the following sub-heads:

Technical Correctness	10 points
(The quality of the design too has to be considered here)	
Testing	12 points
Synthesis	10 points
Back Annotation	8 points
Presentation	10 points
Total	50 points

- 9.3. **Project Via vas:** This will have two sub-heads:

- 9.3.1. **Project presentations by the group:** This has a team wise credit. **(10 points)**

- 9.3.2. **Individual Via vas:** The contribution of the individual in the project has been given a credit. This oral test will have project specific questions to judge the individual understanding of the subject and his contribution **(25 points)**

Total = 35 points

- 9.4. Based on evaluations, final grade will be awarded & communicated to C-DAC ACTS Pune

10. Ensuring Security of Evaluation data/records:

- 10.1. Ensure that all data relating to evaluation of students is stored in a secure place that cannot be accessed by unauthorized personnel.
- 10.2. All question papers must be prepared and stored in a separate area specifically designated for the purpose.
- 10.3. Whenever any external faculty sets a question paper, ensures that he should follows the guidelines given by C-DAC ACTS.
- 10.4. Ensure that only one copy of any question paper is prepared in physical (printed) form for review and revision.
- 10.5. When the question paper is finalized, print out one master copy and gets it signed by the paper setter, Reviewer and DRM Evaluation.
- 10.6. Prepare required number of photocopies of the question paper and store them in a safe and secure location before the exam.
- 10.7. The data relating to evaluation of students, such as soft copies of question papers and answer keys, student marks database and performance statements etc. must be kept in a separate domain/directory which is accessible only to authorized personnel. Ensure that the data is regularly backed up.
- 10.8. The question papers for the theory as well as the laboratory examinations at all the centres will be set by resource Center. The centers according to guidelines provided by resource Center, will conduct the evaluation of the laboratory and assignments locally.

Suggested Teaching Guidelines for **OOPs with C++ Programming – PG-DMC August 2019**

Duration: 30 class room hrs. + 30 lab hrs. (60 hrs.)

Objective: To reinforce knowledge of Object Oriented Concepts and C++ Programming.

Prerequisites: Knowledge of C Programming

Evaluation method: Theory exam– 40% weightage
Lab exam – 40% weightage
Internal exam– 20% weightage

List of Books / Other Training Material

Courseware:

C++ Primer Plus by Stephen Prata / Pearson

Reference:

1. Thinking in C++ by Bruce Eckel
2. The C programming Language by Kernighan and Retchie
3. The C++ Programming Language, Bjarne Stroustrup;
4. Object-oriented Analysis And Design Using Umlan Introduction To Unified Process And Design Patterns 1st Edition by Mahesh P. Matha / PHI
5. Modern C++ Programming with Test-Driven Development: Code Better, Sleep Better

Note: Each session having 2 Hours

Session 1 & 2: (Linux Command and Revision of C Programming)

Lecture

- Linux introduction
- Basic Linux commands required for writing and executing C++ programs
- Revision of C Programming
 - Pointers
 - Functions (Call by value and reference)
 - Recursion
 - Arrays using Pointers
 - Structures
 - Union
 - Enumeration and Typedef
 - File handling

Assignment – Lab:

- Practice of Linux commands
- Writing different C programs based on functions, structures, file and executing them

Session 3: (OOP Concepts)

Lecture

- Difference between C and C++
- Discussion on Object oriented concepts

Suggested Teaching Guidelines for **OOPs with C++ Programming – PG-DMC August 2019**

- Classes and Objects, Access Specifiers, Overloading, Inheritance, Polymorphism
- Constructors and Destructors
- Namespaces
- Templates

Assignment – Lab:

- Write a Student class and use it in your program. Store the data of ten students and display the sorted data according to their roll numbers, date of births, and total marks.

Session 4: Beginning with C++

Lecture

- C++ Tokens
- Initialization
- C++ Operators
- C++ Type Conversion & Type Casting
- Static Members
- Constant Members
- Expressions
- Control Structure

Assignment – Lab:

- Implement all C++ operators
- Write a program to convert a distance entered in Feet and Inches to Meter using class to basic data type conversion.
- Declare members and implement in your programs.
- Implement all control structures through your program

Session 5: Functions in C++

- Different forms of functions
- Function prototyping
- Call by Reference
- Inline Functions
- Math library functions etc.

Assignment – Lab:

- Implement functions through your program
- Declare function and call it by reference and note the observations
- Implement Inline functions in your program

Session 6 & 7: Memory Management and pointers

Lecture

- Introduction to memory management in C++
- Pointers in C++
- Arrays using pointers
- Enumeration
- Typedef

Suggested Teaching Guidelines for **OOPs with C++ Programming – PG-DMC August 2019**

- Using New operator
- Pointers to objects
- this pointer
- Pointers to derived class
- Comparison of new over malloc, calloc and realloc etc.
- Memory freeing using Delete operator

Assignment – Lab:

- Assignments using pointers, arrays of pointers
- Assignments on passing pointers in functions.
- Using pointers write your own functions for the following;
 - String comparison
 - String concatenate
 - String copy
 - String length

Note: Do not include <string.h> in your program and Implement delete operator in your program

Session 8: Constructor and Destructor

Lecture

- Constructors
- Parameterized constructors
- Multiple constructors in class
- Dynamic initialization of objects
- Copy Constructors
- Encapsulating into an object
- Destructors
- Associations
- Inner Classes

Assignment – Lab:

- Implement constructor and destructors through your program
- Write program to implement inner class in C++

Session 9 & 10: Inheritance, Virtual Functions, Polymorphism

Lecture

- Inheritance, Virtual Functions, Polymorphism
- Inheritance
- Type of Inheritances
- Run Time Polymorphism
- Polymorphism
- Overloading functions
- Overloading Operators
- Friend Functions
- Virtual Functions and Pure virtual functions

Assignment – Lab:

- Write Date and Time classes that allows you to add, subtract, read and print simple dates in dd/mm/yyyy and time in hh:mm:ss forms. Use function overloading in your program.

Suggested Teaching Guidelines for **OOPs with C++ Programming – PG-DMC August 2019**

- Assignments to overload =, ==, +, ++, --, <<, >> and [] operators.
- Design a hierarchy of computer printers. Use multiple inheritance in your hierarchy. Also use friend functions and classes in your program.
- Using virtual and pure virtual functions implement hierarchy of computer printers.

Session 11: Inheritance – extending class

Lecture

- Introduction
- Types of inheritance
- Single inheritance
- Multiple inheritance
- Multilevel inheritance
- Hierarchical inheritance
- Hybrid inheritance etc
- Virtual base class
- Interfaces
- Abstract class
- Constructors in derived class

Assignment – Lab:

- Implement Abstract classes in your program

Session 12: Exception Handling

Lecture

- Exception Handling Introduction
- Exception handling – throwing, catching, re-throwing an exception
- Specifying exceptions etc.

Assignment – Lab:

- Implement exceptions and do proper management through your program

Session 13: Managing Console I/O Operations

Lecture

- Introduction
- C++ streams
- C++ stream classes
- Unformatted I/O operations
- Formatted I/O operations
- Managing output with manipulators

Assignment – Lab:

- Implement console I/O operations through your program

Session 14: File Handling in C++

Lecture

- Definition of file
- File handling in C++
- Doing read, write operation in files

Suggested Teaching Guidelines for
OOPs with C++ Programming – PG-DMC August 2019

Assignment – Lab:

- Assignments on files doing different operations.

Session 15: Test-driven development and unit testing with examples in C++

Lecture

- Basic concepts of Test-driven development & Unit testing with examples in C++

Suggested Teaching Guidelines for **Algorithm and Data Structures – PG-DMC August 2019**

Duration: 30 class room hrs. + 30 lab hrs. (60 hrs.)

Objective: To reinforce knowledge of Problem solving techniques, Data Structure Concepts and analysis of different Algorithm.

Prerequisites: Knowledge of Programming in C/C++ with Object Oriented concepts

Evaluation method: Theory exam– 40% weightage
Lab exam – 40% weightage
Internal exam– 20% weightage

List of Books / Other training material

Courseware:

Fundamentals of Data Structure In C++, Horowitz, Sahni Sartaj

Reference:

1. Problem Solving: Best Strategies to Decision Making, Critical Thinking and Positive Thinking by Thomas Richards / Kindle Edition
2. Fundamentals of Data Structures in C++ by Horowitz, Sahani & Mehata / Orient Longman
3. Data Structures, Algorithms and Applications in C++ by Sartaj Sahni
4. Object-oriented Analysis And Design Using Umlan Introduction To Unified Process And Design Patterns 1st Edition by Mahesh P. Matha / PHI
5. Introduction to Algorithms by Cormen, Leiserson, Rivest and Stein
6. Data Structures Via C++ Objects by Evolution by A Michael Berman / Oxford University Press
7. Design and Analysis of Algorithm by S Sridhar / Oxford University Press
8. Fundamentals of Computer Algorithms 2nd Edition by Sanguthevar Rajasekaran, Sartaj Sahni, Ellis Horowitz / Orient Longman
9. Introduction to Algorithms by Al. Cormen / PHI Learning
10. The Algorithm Design Manual by Steven S. Skiena / Springer
11. Algorithmic Puzzles by Anany Levitin, Maria Levitin / Oxford University Press

Note: Each session having 2 Hours

Session 1: Introduction to STL

Lecture

- Object Design and Templates
- STL (Standard Type Libraries)

Assignment – Lab:

- Find the number of students who are passes or failed using MAP.
- Find the prime numbers from 2 to n using sieves algorithm, use SET

Suggested Teaching Guidelines for **Algorithm and Data Structures – PG-DMC August 2019**

Session 2,3: Problem Solving & Computational Thinking

Lecture

- Define the problem
 - Identify the problem
 - Introduction to Problem Solving
 - Problem solving basics
 - Defining creativity v/s innovation
- Find Creative Solutions using creativity tools
 - Effective problem solving approaches
 - Critical thinking and information analysis
 - Brainstorming, Reverse Brainstorming, Imagineering, Mind Mapping,
 - Six Thinking Hats: A Tool to Strengthen Critical Thinking, Collaboration, Communication, and Creativity Skills
 - Analyzing the situation, Gathering information, Identifying solution criteria
 - Decision Making Methods
 - Charts and Diagrams
 - Applying outcome-based thinking
- Evaluate and Select solution
 - Pro's and Con's, Force field analysis, Feasibility/Capability Analysis,
 - Decision analysis, evaluating problems
 - Choosing among alternatives
 - Qualitative analysis, discussing qualitative analysis techniques
 - Establishing objectives
 - Assigning weight to objectives in order to make the best decision
 - Creating a satisfaction scale to choose between alternatives
- Implementing Decisions
 - Create an action plan
 - Break solution into action steps
 - Prioritize actions and assign roles (setting priorities for taking action)
 - Follow-up at milestones

Assignment – Lab:

- Faculties need to assign different problems, mostly real world problems
- Students (by team wise, there are two students in a team) need to analyze as per the techniques learned
- Students need to solve by the thinking approaches learned.
- Based on the above problems students need to select as per the selection criteria learned
- They need to implement the selected solution and need to do the documentations.

Session 4, 5: Algorithm design

Lecture

- How to write efficient Algorithm
- Introduction to algorithm design techniques
- Algorithm Design techniques
- Analysis of an Algorithm

Suggested Teaching Guidelines for
Algorithm and Data Structures – PG-DMC August 2019

- Asymptotic analysis
- Algorithm analysis
- Analysis of different type of Algorithms
 - Divide and Conquer Algorithm
 - Greedy Algorithm
 - Dynamic Programming Algorithm
 - Brute force Algorithm
 - Backtracking algorithms
 - Branch-and-bound algorithms
 - Stochastic algorithms
- Complexity
 - Complexity Analysis
 - Space complexity of algorithm
 - Time complexity of algorithm
- Case study on Algorithm Design techniques
- Application of Data structures

Assignment – Read:

- Study on different Algorithms
- Compare different Algorithms previously programmed and do the analysis

Session 6, 7: Algorithm & Data Structures

Lecture:

- Introductory Concepts
- Algorithm Constructs
- OO design: Abstract Data Types (ADTs)
- Basic Data Structures
 - Arrays
 - Stacks
 - Queues
 - Circular Queues
 - Priority Queues
 - Deques

Assignment – Lab:

- Implement Stack through Array
- Implement C-Stack, C2-Stack and CN-Stack in same memory block.
- Implement Queues with inserting element at different location (First, Last and at specific location)
- Implement circular queue, Priority Queues and Dqueue
- Implement program to convert infix expression into postfix expression & evaluate postfix expression.

Session 8,9: Basic Data Structures (Cont...)

Lecture

- Linked lists

Suggested Teaching Guidelines for
Algorithm and Data Structures – PG-DMC August 2019

- Single Linked Lists
- Double Linked Lists
- Circular Linked Lists
- Node-based storage with arrays

Assignment – Lab:

- Implement circular queue using linked list
- Design an iterator using circular linked list

Session 10,11: Trees & Applications

Lecture

- Introduction to trees
- Trees and Terminology
- Tree traversals
- Ordered trees
- Binary trees
- Complete binary trees
- Search trees
- Binary search trees
- Introduction to self balancing tree & variants

Assignment – Lab:

- Write a program to implement a binary search tree and the following operations on it:
 - Create()
 - InsertNode()
 - Tree traversals (Inorder(), Preorder(), Postorder())
 - deleteNode()
- Design a threaded binary tree and implement the orders.

Session 12: Searching & Sorting algorithms

Lecture

- Objectives of Searching
 - The Sequential Search
 - Analysis of Sequential Search
 - The Binary Search
- Analysis of Binary Search
- Introduction to sorting
 - Selection sort
 - Insertion sort
 - Bubble sort
 - Heap sort
 - Merge sort
 - Quick sort
- Analysis of sorting algorithms

Assignment – Lab:

- Writing program to search an item through sequential search technique.

Suggested Teaching Guidelines for **Algorithm and Data Structures – PG-DMC August 2019**

- Implement to find an item in a list through binary search
- Implement sorting algorithm for selection sort, Bubble sort, heap sort and quick sort
- Write a program to merge two sorted linked lists

Session 13: Hash functions and hash tables

Lecture

- Hashing & Introduction to hash tables
- Hash functions
- Mapping down to $0 \dots M - 1$
- Chained hash tables
- Scatter tables
- Open addressing
- Linear probing
- Quadratic probing
- Double hashing
- Poisson distribution
- Collision Resolution
- Analysis of Hashing

Assignment – Lab:

- Implement hashing techniques in different programs solved earlier
- Implement collision and solution to it on any previous solved problem

Session 14,15: Graph & Applications

Lecture

- Introduction to graph theory
- Graph Terminology
- Different types of Graphs
- Representation of Graphs
 - Connectedness, Single source un-weighted path length, identifying bipartite graphs
 - Graph Traversal Algorithms (Breadth First Search, Depth First Search)
 - Single-source shortest path algorithms, Dijkstra's algorithm, A* search algorithm, Bellman-Ford algorithm
 - All-pairs shortest path, Floyd-Warshall algorithm, Johnson's algorithm
 - Maximum flow algorithms, Ford-Fulkerson algorithms
- Spanning Trees
 - Minimum spanning tree algorithms, Prim's algorithm, Kruskal's algorithm

Assignment – Lab:

- Implement a graph using adjacency links and traverse using Depth First Search.
- Write a program using STL to implement Dijkstra's Shortest Path Algorithm.

Suggested Teaching Guidelines for

OS Concepts and Linux programming– PG-DMC August 2019

Duration: 20 Class Room hrs. + 20 Labs hrs. **(40 hrs.)**

Objective: How to effective application on OS and Linux used in mobile phones and handheld devices

Prerequisites: Familiar with programming concepts and operating system basics.

Evaluation method: Theory exam – 40% weightage
Lab exam – 40% weightage
Assignments – 20% weightage

List of Books / Other training material

Courseware:

1. Beginning Linux Programming by Richard Stones, Neil Matthew / Wiley
2. Operating Systems Concepts by Galvin / Wiley

Reference Books:

1. Beginning Linux Programming by Richard Stones, Neil Matthew (Wiley Dreamtech)
2. Operating Systems Concepts by Silberschatz, Galvin

Note: Each session having 2 Hours

Session 1:

- Introduction to Operating System
- History of operating System in brief
- System Components/Services
- Interrupts & System calls

Assignment – Reading:

Read and understand System Structure and Process Scheduling

Session 2: Process - architecture

- Introduction to Process Management and related data-structures
- Process states and life cycle
- Multithreading

Session 3: CPU Scheduling

- Basic concepts
- Scheduling criteria
- Scheduling Algorithms
- **Linux scheduling policies – normal vs real-time scheduling and priorities**

Suggested Teaching Guidelines for

OS Concepts and Linux programming– PG-DMC August 2019

Session 4: Process Synchronization

- Critical –section Problem
- Critical Region
- Semaphores & Mutex
- Producer Consumer problem
- Monitors
- Deadlocks
 - Deadlock Characterization
 - Deadlock Prevention
 - Recovery techniques

Assignment – Reading:

Read and understand Deadlock Detection, Recovery from Deadlocks

Session 5, 6&7:

- Introduction to memory management
 - Logical Address
 - Physical Address
 - Dynamic linking
- Memory Management
 - Contiguous Allocation
 - Segmentation
 - Paging
 - Segmentation with paging
- Virtual Memory
 - Swapping
 - Demand Paging
 - Page fault exception handler
 - Page-Replacement Algorithms
 - Thrashing

Assignment – Reading:

Read and understand File-System Structure

Session 8:

- File System Interface and Implementation
 - Files and Directories Operations
 - File System mounting and protection methods
 - File System Structures and Implementation
- Secondary Storage Structure
 - Disk Structure
 - Disk Scheduling and Management
 - Swap-Space Management

Suggested Teaching Guidelines for

OS Concepts and Linux programming– PG-DMC August 2019

Session 9 & 10:

- What is a shell?
- Different types of Linux shells
- Bourne Again Shell (BASH)
- Shell variables (environment and user defined)
- Shell files (.bashrc, .profile, .bash_profile, .bash_logout)
- Positional parameters
- Wild cards (* and?)
- Command line arguments
- Arithmetic in shell scripts
- *read* and *echo* commands in shell scripts
- The *tput* command
- Taking decisions:
 - *if-then-fi*
 - *if-then-else-fi*
 - The *test* command (file tests, string tests)
 - Nested *if-elses*
 - The *case* control structure
- The loop control structures
 - The *while*, *until* and *for* loop structures
 - The *break* and *continue* statements
- Shell metacharacters
- Command line expansion
- Directory stack manipulation
- Job control, history and processes
- Built-ins and functions

Assignment –Lab:

Different Assignments on shell scripting will be provided by centers

Suggested Teaching Guidelines for **Introduction to DBMS – PG-DMC August 2019**

Duration: 20 classroom hrs. + 20 lab hrs. **(40 hrs.)**

Objective: To introduce the student to DBMS and Facilitate Hands On Experience on different kinds of Databases.

Prerequisites: Fundamentals of Computer and any programming language

Evaluation method:

Theory exam–	40% weightage
Lab exam –	40% weightage
Assignments –	20% weightage

List of Books / Other training material

Courseware:

Textbook of RDBMS by [Mrs Vidya H Bankar](#)

Reference:

1. MySQL By DuBois, Pearson
2. Mastering Database Technologies by Ivan Bayross / BPB
3. Database Management Systems by Gerald V Post
4. MongoDB: The Definitive Guide by Kristina Chodorow and Michael Dirolf

Note: Each session having 2 Hours

Session 1:

Lecture

- Introduction to DBMS – What is DBMS, Its need
- Areas where DBMS are used
- Introduction to RDBMS
- Codd's 12 rules for a Relational Database (conclusion)
- Need for Normalization.

Assignment – Reading:

Read and understand Chapters for the next session

Session 2:

Lecture:

- Various normalization forms 1st normal form, 2nd normal form
- 3rd normal form,
- Introduction to 4th, BCNF, etc
- Need for Denormalization

Assignment – Reading:

Read and understand Chapters for the next session

Suggested Teaching Guidelines for **Introduction to DBMS – PG-DMC August 2019**

Session 3:

Lecture:

- DDL Commands
- DML & DCL Commands

Assignment – Lab:

SQL Practice Questions Including:

DDL Commands: Create/Alter/Drop/Grant/Revoke

DML Commands: Select/Insert/Update/Delete/Truncate

DCL Commands: RollBack/Commit

Assignment – Reading:

Read and understand Chapters for the next session

Session 4:

Lecture:

- Inbuilt Functions
- Grouping Things Together (Group By, Having Clause)
- Set Operators (UNION, UNION ALL, INTERSECT, MINUS)
- Subqueries
- Joins

Assignment – Lab:

Number Functions: -

Single Value Functions: NVL, ABS, CEIL etc

Group Value Functions: AVG, COUNT, MAX etc

SQL Practice Questions:-

Queries containing Group By, Having Clause and set operations

Correlated Queries, SubQueries, Outer Joins

Session 5 & 6:

Lecture:

- Introduction to MYSQL Programming
- Exceptions

Assignment – Lab:

SQL Practice Questions covering MYSQL Blocks & Exceptions

Assignment – Reading:

Read and understand Chapters for the next session

Suggested Teaching Guidelines for
Introduction to DBMS – PG-DMC August 2019

Session 7 & 8:

Lecture:

- Procedures, Trigger
- Functions, indexes

Assignment – Lab:

SQL Practice Questions covering Trigger & Procedures.

SQL Practice Questions covering Functions and Indexes

Assignment – Reading:

Read and understand Chapters for the next session

Session 9 & 10:

Lecture:

- Introduction to No SQL
- NoSQL Database Types
- Introduction to Mongo DB & It's benefits
- Introduction of Dynamic Schemas & Auto Sharding concepts in Mongo DB

Assignment – Lab:

Practice CRUD operations in Mongo DB

Suggested Teaching Guidelines for **Java Programming I (Core Java) – PG-DMC August 2019**

Duration: 34 class room hrs. + 36 lab hrs. **(70 hrs.)**

Objective: To introduce the student to Core Java

Prerequisites: Knowledge of object-oriented programming

Evaluation method:

Theory exam –	40% weightage
Lab exam –	40% weightage
Assignments –	20% weightage

List of Books / other training material

Courseware:

Core and Advanced Java Black Book by Dreamtech Press

Reference:

1. Beginning Java 2 by Ivor Horton; Wrox Publication

Note: Each session having 2 Hours

Session 1

Lecture

- Java Language and its features
- How Java is different from C++?
- JDK , JRE & JVM
- JVM Architecture Overview
- The first Java Program
- The Java Language –
- Data types, Variables, Constants, operators, Control Statements (if, for, while, switch etc.)
- Classes in java
- Constructors, instance data and methods, the new operator

Assignment –

- Create Java Program for simple calculator , compile & test it.
- Create Emp class with its suitable data members. Create Tester class to create Emp instance & understand memory picture.

Assignment – Tutorial:

- Compare syntactical similarities and dissimilarities between Java and C++

Session 2

- Methods, overloading, parameter passing, objects as parameters
- Memory management, garbage collection
- The this facility, static data and methods, block, scope, lifetime
- JDK tools (Java Compiler, Java Runtime, Java Debugger, Java doc)

Suggested Teaching Guidelines for
Java Programming I (Core Java) – PG-DMC August 2019

Assignment – Reading:

Refer to JVM specifications to understand Garbage collection.

Refer to Java tutorial for instance & static methods.

Session 3:**Lecture**

- Packages
- Access Control Rules
- Types of Inheritance
- IS A Relationship
- Inheritance & Polymorphism

Assignment – Lab:

Create Emp based organization structure having suitable sub types & tester classes.

Tester classes & core organization classes must be in separate packages.

Session 4

- Compile Time Polymorphism
- Run time Polymorphism (Dynamic method Dispatch)
- Abstract Classes
- Final keyword
- Interfaces
- Dynamic method dispatch using Interfaces.

Assignment – Lab:

Implement Fixed & Growable stack to demonstrate Java interface & implementation classes.

Session 5:**Lecture**

- Inner classes
- Exception Handling
- Exception class inheritance hierarchy
- Try, catch , finally , throw , throws
- Custom exception classes

Assignment – Lab:

Create custom exception class for supporting validation rules in customer scenario.

Session 6:**Lecture**

- The java. Lang package, Object, Number, Math, System
- The String class
- Wrapper classes
- The java.util Package
- ArrayList,LinkedList,Vector,HashSet,LinkedHashSet,TreeSet,HashMap,TreeMap

Suggested Teaching Guidelines for Java Programming I (Core Java) – PG-DMC August 2019

Assignment – Lab:

Implement Customer based data structure to perform CRUD operations with linked list using collection's framework and without using collection's framework.

Note: Design algorithms using pseudo code before implementation

Design algorithms using pseudo code for following using arrays. Also implement the same.

- Searching: Linear Search
- Sorting: Bubble sort

Session 7,8,9:

Lecture

- Introduction to Generics
- Generic classes & interfaces .
- Wildcard syntax
- Inheritance in Generics
- The Java2 Collection Framework
- List , Set & Map Functionality
- Sorting using Natural Ordering & Custom Ordering

Assignment – Lab:

Refer to Java tutorial's generics trail.

Design emp based organization for supporting CRUD + Sorting functionality.

Must be designed in suitable layers to emphasize separation

Session 10,11:

Lecture

- The java.io Package
- Byte Streams and Unicode Character Streams
- File Handling
- BufferedReader,PrintWriter streams
- Persistence of objects
- Object Serialization & de serialization

Assignment – Lab:

- Make your above Employee, manger classes objects persistent.

Session 12,13:

- Multithreaded programming in Java
- Multitasking --Process Based Vs Thread Based
- Thread state transitions
- The Thread class & its API
- The Runnable interface
- Thread synchronization techniques
- Applying thread safety to Collection framework classes.

Suggested Teaching Guidelines for
Java Programming I (Core Java) – PG-DMC August 2019

Assignment – Lab(3 hours):

- Apply multi threading techniques to File handling & ensure thread safety.
- Using thread safe vs thread un safe collections.

Session 14,15:

Lecture

- Introduction to Java Network Programming
- Inet Address
- Socket, and ServerSocket
- Attaching suitable data streams & data transfer.
- Developing multi threaded server for handling concurrent clients.

Lecture

- Reflection Overview
- Annotation Overview

Assignment – Lab:

- Write a multithreaded chat server and client.

Session 16,17

Lecture

- Database Access Methods , JDBC, driver & architecture
- The java.sql package
- Driver Manager, Connection, Statement, PreparedStatement, Result Set
- CallableStatement ,Stored procedure & functions
- Scrollable ResultSet
- Writing multi tiered DB applications (Use DAO & DTO layers)
- Handling BLOB to store & retrieve binary content

Assignment – Lab:

Build an application to get student's details using database.
Store & retrieve images from DB.
Invoke stored procedure & a function

Suggested Teaching Guidelines for

Java Programming II (Advanced Java)– PG-DMC August 2019

Duration: 40 hrs. class room + 40 hrs. lab (80 hrs.)

Objective: To introduce the student to Core Java & Advanced Java

Prerequisites: Knowledge of object-oriented programming

Evaluation method: Theory exam – 40% weightage
Lab exam – 40% weightage
Assignments – 20% weightage

List of Books / other training material

Courseware:

Core and Advanced Java Black Book by Dreamtech Press

Reference:

Advanced Java Programming by Uttam Roy

Note: Each session having 2 Hours

Session 1:

Lecture

Introduction to JSON

- Advantages of JSON
- Parsing API
- Java to JSON & JSON to Java conversion
- Handling JSON types.

Assignment – Lab:

- Create Java Application to test JSON API

Session 2:

Lecture

- Servlets: Dynamic Content Generation
- Advantages of Servlets over CGI
- The Servlet interface
- The HttpServlet, HttpServletRequest, HttpServletResponse classes
- Exception Handling & Integration with DB using J2EE layers (DAO & DTO)

Session 3:

Lecture

- Session
- Session Management
- Session Tracking with
 - Cookies
 - HttpSession API

Suggested Teaching Guidelines for

Java Programming II (Advanced Java)– PG-DMC August 2019

- Page Navigation Techniques
 - Send Redirect
 - Request Dispatcher

Assignment – Lab:

- Use Java Servlets technology in designing and implementing an online shopping cart. Incorporate Sessions in the same.

Assignment – Reading:

- Know more about the HTTP protocol at www.w3c.org

Assignment – Tutorial:

- Compare which way of session tracking is better Cookies or HttpSession

Session 4:

Lecture

- JSP: Separating UI from Content generation code
- MVC architecture
- Life cycle of a JSP page
- Directives, Implicit Objects, Scriptlets, Expressions & Expression Language

Assignment – Lab:

- Separate UI code from the controller code in your Online shopping cart by incorporating JSP and Servlets

Session 5:

Lecture

- Scope
- Java Beans
- JSP Using Java Beans actions
- Session Tracking, JSP standard Actions
- Java Server Pages Standard Tag Library (JSTL)
- Design a complete web application using JSP-Java Bean -DAO -DTO Layers

Assignment – Lab:

- Complete the implementation Online Shopping cart.

Session 6,7:

Lecture

Introduction to Hibernate Framework

- ORM Concept
- JPA
- Hibernate Architecture
- Configuration details
- SessionFactory & Hibernate Session
- Hibernate annotation support using JPA
- Hibernate ID generators

Suggested Teaching Guidelines for

Java Programming II (Advanced Java)– PG-DMC August 2019

- DB Dialects
- Creating layered Hibernate based Java SE application (Hibernate Based DAO + JPA Entities)

Assignment – Lab:

Create from scratch, hibernate based Java SE application containing layers
Tester --DAO --DTO--DB

Assignment – Reading:

Refer to Java Persistence Book to understand ORM & hibernate's advantages over JDBC.

Session 8:

- DB Transaction management
- Hibernate Session CRUD API
- openSession vs getCurrentSession
- Entity Life cycle consisting of Transient, Persistent , Detached & Removed states.

Assignment – Lab:

- Create Hibernate based Java SE application for bank scenario.

Session 9,10:

Advanced Hibernate Topics

- Hibernate Caching
- HQL
- Association between Entities
 - One -One
 - One -Many
 - Many-One
 - Many-Many

Entity Types Vs Value Types

Mapping between Entity & Value Types

Criteria Queries

Named Queries

Assignment – Lab:

Use Book & Author scenario for all types of Entity --Entity associations.

Use Author & AuthorDetails scenerion to test the mapping between Entity & Value type.

Session 11:

Lecture

Spring Framework Overview

- Why Spring
- Overview of Spring 4 Modules
- Inversion Of Control (IoC)
- Dependency Injection
- Spring Beans life cycle

Suggested Teaching Guidelines for

Java Programming II (Advanced Java)– PG-DMC August 2019

- Scopes of spring beans
- XML Based configuration
- Creating Spring Based Java SE Application in Eclipse using STS plugin

Assignment – Lab:

- Create Spring based Application to test dependency injection & loose coupling of objects.

Session 12,13:

Lecture

Spring Core Container

- Setter Based Dependency Injection
- Constructor Based DI
- Factory Method Based DI
- Combining DI Techniques
- Autowiring vs Explicit wiring
- Autowiring By Name & Type
- AutoWiring by constructor
- Collection handling
- Annotation Based configuration

Assignment – Lab:

Create Message sender application to test loose coupling & dependency Injection.

Initially implement it using XML based configuration. Later reduce XML content & replace it by annotations.

Session 14,15:

- MVC Architecture
- Implement MVC based web application using Servlet,JSP & Java Bean

Spring Web MVC

- Spring MVC Flow
- HandlerMapping , ViewResolver & DispatcherServlet
- Spring Controller Beans
- RequestMapping annotation
- RequestHandling Methods
- Conversion & Validation
- Complete web application using Controller, Service & DTO layers.

Assignment – Lab:

- Create Spring MVC web Application for Contact Manager application.

*Suggested Teaching Guidelines for***Java Programming II (Advanced Java)– PG-DMC August 2019****Session 16,17****Lecture****Spring Hibernate Integration**

- Spring Hibernate Configuration setup.
- DataSource Bean, SessionFactory Bean & TransactionManager Bean
- Hibernate based DAO layer.
- Spring Managed Service Layer to handle transactions.
- Integration & Testing.

Assignment – Lab:

- Implement airline reservation system by integrating spring MVC & hibernate based persistence layer.

Session 18,19,20:

- Introduction to Web Services
- SOA basics
- SOAP based Vs RESTful web services
- REST Architecture
- Introduction to Spring Boot
- Building Restful Web Services with Spring Boot
- Content Negotiation
- Implementing Static & Dynamic Filtering
- Versioning RESTful Services: Basic and Header & Content Negotiation Approach
- Introduction to JPA
- Connecting RESTful Web Services to JPA
- Integration & Testing of RESTful server & client
- Introduction to Java Mail
- Setting up Java mail using Spring framework
- Sending Mail.

Assignment Reading

Refer to J2EE tutorial to understand, need & implementation of web services.

Assignment - Lab:

- Develop a payment gateway based upon RESTful web service for any e-commerce based application viz Online shopping.
- Create spring MVC based web application to send mail to the recipient.

Suggested Teaching Guidelines for
iOS Programming – PG-DMC August 2019

Duration: 58 hrs. Class room + 56 hrs. Lab **(114 hrs.)**

Objective: How to effectively build an application on iOS used in mobile phones and handheld devices

Prerequisites: Familiar with programming concepts and operating system basics. Students should be proficient in object-oriented programming using C++ before this module

Evaluation method: Theory exam– 40% weightage

Lab exam – 40% weightage

Assignments – 20% weightage

List of Books / other training material

Courseware:

Professional iOS Programming with Swift by Michael Dippery

Reference:

1. Mastering iOS 10 Programming by Donny Wals
2. Head First iPhone Development by Dan Pilone / O'reilly
3. The Swift Programming Language Documentation

Language of teaching – Swift 3.0 or later

Xcode – version 8.0 or above

Note: Each session having 2 Hours

Session 1, Session 2

- The iOS Technology Stack
 - About iOS Technology Overview
 - The Core OS Layer
 - The Core Services Layer
 - The Media Layer
 - The Cocoa touch Layer
- Getting familiar with XCode
 - Storyboard
 - Various features of XCode
- Basic Data types in Swift
 - Integers
 - Floating point numbers
 - Booleans
 - Numeric Literals
 - Integer Literals
 - Float Literals
- Constants and Variables
 - var

Suggested Teaching Guidelines for
iOS Programming – PG-DMC August 2019

- let
- Type Annotation
- Typecasting in Swift
 - Any basic type to another basic type
 - String to any basic type
- Math functions in Swift
 - Arithmetic operators and compound assignment.
 - Increment and Decrement operators.
- Conditionals
 - if..else
 - Nested if...
 - else..if
 - Ternary Operator
- Switch case and range
 - Loops
 - for loop
 - for in loop
 - while loop
 - repeat..while loop

Assignments - See file attached

Session 3,4

- Optional Operator
 - Understanding the optional operator - "?"
 - Various uses of optional operator
 - Defining an optional
 - Safely accessing methods or properties
 - Safe conditional conversion
 - Nil coalescing operator - ??
- Typecasting using "as" operator
 - Guaranteed conversion - as
 - Conditional conversion - as?
 - Forced conversion - as!
- "!" Operator
 - Obtaining value of an optional variable
 - Defining variables that are initially nil but get set soon afterwards, And are guaranteed not to be nil after that (called implicitly unwrapped Optionals).
- Strings and Characters
 - Strings and String operations - Concat, Append, Interpolation, count characters, access string from index, insert and remove, String compare, has suffix and prefix.

Suggested Teaching Guidelines for
iOS Programming – PG-DMC August 2019

- Characters and character operations
- Tuples in Swift
 - Understanding a tuple
 - Tuple operations - access and edit.

Assignments - See file attached

Session 5,6

- Collections in Swift
 - Arrays
 - Creating an array
 - Array Operations
 - Dictionaries
 - Creating a dictionary and inserting an element into it
 - Dictionary operations
 - Sets
 - Creating a set
 - Set Operations
- Functions in Swift
 - Different types of functions
 - Function with multiple return values.
- Classes and Structures in Swift
 - Basic class and structure
 - Difference between class and Structure
- Initialization and Deinitialization
- Inheritance
 - Overridden and overriding functions
 - Super and dot operator
- Closures and Enumerations
- Properties
 - Stored properties
 - Compound properties
 - Property observer
- Methods
 - Instance method
 - Self property
 - Type method
- Protocols and Extensions
- Introduction to generics

Suggested Teaching Guidelines for
iOS Programming – PG-DMC August 2019

Assignments - See file attached

Session 7,8

- Basic components of iOS SDK
 - Storyboard - Interface builder
 - ViewController - Life Cycle of a view controller
 - AppDelegate - Life cycle of app through app delegate
 - Code Debugger
 - MVC design strategy
 - MVVM design strategy
- Hello World
 - Anatomy of a Hello World Project
- Single Page applications
 - Labels and TextField
 - Buttons
 - ImageView
 - Sliders and Switches

Assignments

- 1.Create a Basic Calculator app
- 2.Create an app to check if a number is prime
- 3.Create an app for controlling volume using slider
- 4.Create an app where, the text of label changes when state of switch changes.

Session 9,10

- Concept of Bounds & Frame; Their differences
- Concept of Constraints and AutoLayout
- AutoLayout using Interface Builder
- AutoLayout completely from code
- Uses of UIStackView in making adaptive layout
-

assignments:

- 1.Create an app using some UI elements and make the autolayout using code

Session 11,12

- NavBar and Navbar Button items
- Toolbar and Toolbar Button items
- Timers
- Segues and Navigation Controller
- Container Views
- Advanced Segues.
- Observer & notification

Suggested Teaching Guidelines for
iOS Programming – PG-DMC August 2019

Assignments

1. Create a Basic Stopwatch app
2. Create a Basic User register, login flow using multiple view controllers
3. Perform changes in one ViewController from another based on some condition using observer and notification.

Session 13,14

- Alert Controller
- Alert Sheet
- Activity Indicator
- Progress Views
- Segmented Control
- Search Bar
- Modals
- Popovers
- Date Pickers
- Custom Picker

Assignments

1. Create a simple app to fill a form, show an alert if an item is unfilled
2. Create an app which shows progress from 0 to 100 using progress view.
3. Create an app which demonstrates – Modals, Popovers, Date picker, Custom picker

Session 15,16

- TableView with default cell
 - Implementing TableView with default cell style
 - Various behaviours in TableView
- Table View with custom & reusable cell
 - Implementing TableView with UITableViewCell and custom cell style
- Scroll View
 - Implementing Scroll View
 - Various behaviours in ScrollView
- Collection View
 - Implementing using UICollectionView and UICollectionViewCell classes
 - Various behaviour in collection view

Assignments

1. Create a multiplication table app, such that when the slider is moved to a position, the multiplication table corresponding to that number must be shown in TableView.
2. Create a Recipe book app, on the first page of the app, all the items are shown in a TableView, where each cell of the TableView contains – Image, Label, and Rating bar. When any cell of the TableView is clicked, a new ViewController opens to display the image and recipe of the item clicked

Suggested Teaching Guidelines for

iOS Programming – PG-DMC August 2019

3. Create an Image gallery app using ScrollView, such that different images must be displayed while scrolling back and forth.
4. Create an Image Gallery app using CollectionView.

Session 17,18

- Core Data framework for storing persistent data
 - Create a model
 - CRUD Operations
- Persistent storage using User Defaults
 - Insert, retrieve and delete from UserDefaults.
- Multi threading in iOS
- Retain cycle, ARC, memory leaks
- Tab based application with multiple ViewControllers.
- Page View Controller

Assignments

1. Implement the user login flow by storing the user and login details in Core Data.
2. Create a To Do List app which is Tab based and stores notes in UserDefaults/Core Data.
3. Create an app which runs a timer using Grand Dispatch.

Session19,20

- Core Location and Map Kit
- Using Google Maps in iOS.
- Games and Sprite Kit
 - Sprite and Animation
 - Touch and Gesture events: Pan, Tap, Pinch, Swipe and Shake

Assignments

1. Create an app that shows the current location on Apple maps.
2. Create an app which shows the address of the place where mark is placed.
3. Create an app to demonstrate Animation capabilities in iOS.
4. Create an app in such that the image in image view changes whenever it is swiped in any direction.

Session 21,22

- Playing Audio and Video.
- 3D Touch: Peek and View.
- 3D Touch: Quick View

Suggested Teaching Guidelines for
iOS Programming – PG-DMC August 2019

- Sensors in iOS
 - Core Motion Kit: Accelerometer, Gyroscope etc.
 - Vibration sensors
 - Wi-Fi sensor
- Intro to using Touch ID

Assignments

1. Create a Simple Audio Player app.
2. Create an app which demonstrates the capability of Peek and View, Quick View (Requires iOS device).
3. Create an app which displays the direction of device in degrees and the acceleration of the device in each axis. (Requires iOS device).
4. Create an app such that the phone buzzes when touched on an UIImageView. (Requires iOS device).

Session 23,24

- WebKit/WbView to display web contents
- JSON Parsing
- Intro to using 3rd party APIs in iOS app.
- Introduction to Carthage and Cocoapods
- Use of Some Popular libraries (Alamofire, SwiftyJSON, KeychainSwift etc)
- Using Social media
 - Facebook Login
 - Google Login via Firebase
 - Twitter Login

Assignments

1. Create a Simple app which parses and displays information in a JSON file.
2. Create an app which allows to login on Facebook, Google and Twitter.
3. Create an app to show a website and a custom html in a WebKitWebView.

Session 25,26

- Push Notifications into iOS
 - Showing simple notifications
 - Push Notifications from BaaS (Firebase, Parse etc).
 - Sending push notifications from custom server

Assignments

Create an end-to-end solution to demonstrate push notifications in iOS through – Firebase and custom server.

Suggested Teaching Guidelines for iOS Programming – PG-DMC August 2019

Session 27 & 28

- Introduction to Reactive Programming
 - Advantages of Reactive Programming
 - Integrating RxSwift/RxCocoa in your Xcode Project
 - RxSwift Programming few examples

Assignments

1. Write a program to understand the life cycle of Observable (Next-Error-Completed).
2. Write a simple program to understand the concept of Dispose Bag.
3. Write a program to observe changes to a UIControl & device orientation.
4. Write a simple program to understand the use of Schedulers.
5. Write a simple program to understand the concept of Filtering, Transforming and combining Operators.
6. Write a reactive program to create custom wrappers around existing UIKit components.

Session 29

- Newly added features of Swift 4
- Intro to version control using Git/GitHub from Xcode
- Use of SourceTree
- Overview of app uploading process on Apple Play store.

Annexure:

Assignment for Session 1 – Swift Basics

Note: Write the solution of these problems using Swift language

1. Write a recursive function that will generate and print first n Fibonacci numbers
2. Write a program to print the Pascal triangle for 10 rows.
3. Write a program that will calculate the factorial of any number n.
4. Write a program that will determine if a number is prime or not.
5. Write a program to determine and print the sum of the following harmonic series for a given value if n

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$
6. Write a program that will calculate the sum of the arithmetic progression – $1 + 2 + 3 + \dots + n$.
7. Write a program that converts integer Fahrenheit temperatures from 0 to 212 degrees to floating-point Celsius temperatures with 3 digits of precision. Use the formula

$$\text{celsius} = 5.0 / 9.0 * (\text{fahrenheit} - 32)$$
8. Write a program to validate the Boolean logic – AND, NAND, XOR, NOR, OR , for any 2 given Boolean inputs.
9. Write a program that will calculate and display the real roots of the quadratic equation

$$ax^2 + bx + c = 0$$

Suggested Teaching Guidelines for iOS Programming – PG-DMC August 2019

Using the well know quadratic formula

$$x = (-b \pm \sqrt{b^2 - 4ac}) / 2a$$

Use the function in the program.

10. Print the number of days in a week using Switch case
11. Create a simple calculator program using Switch case

Assignment for Session 2 – Swift Basics

Note: Write the solution of these problems using Swift language

Arrays

1. Given an array of sorted list of integer numbers, write a function to search for a particular item using the method of binary search.
2. Write a program for -
 - a. Transpose of a matrix.
 - b. Determinant of a matrix
 - c. Multiply 2 matrices
3. Write a program, where there are 2 arrays given - arr1 and arr2 having random elements -
 - a. Check if both the arrays are equal
 - b. Check which array is greater of the 2 (based on sum of the elements of each array)
 - c. Determine the highest element from each array
 - d. Calculate the average of the arrays (by adding both the arrays)
4. Given an array of 10 elements -
 - a. Sort the elements of array using Bubble Sort
 - b. Sort the elements of the array using insertion sort
5. Write a program for these operations on an array
 - a. Create an empty array
 - b. Insert any 5 elements into that array
 - c. Change the value of the element at position 3
 - d. Replace the elements from position 2-4 with elements at position 0-1
 - e. Append a new element at the end of the array

Strings and Character

6. Write a program to extract a portion of a character string and print the extracted string. Assume that m characters are extracted, starting with nth character.
7. Write a program that will replace the occurrence of a string **from** with the string **to** in a given input string.
8. In a given long string - The quick fox jumped over the lazy dog
 - a. Calculate the number of vowels (a,e,i,o,u)
 - b. Calculate the number of white spaces
 - c. Generate a new string where 1st letter of each word is in caps.
 - d. Calculate the length of the string
9. For a given quote - Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Suggested Teaching Guidelines for
iOS Programming – PG-DMC August 2019

- a. Delete the first and last characters
 - b. Remove all the words whose word length in 3 or less
 - c. Remove the character at position 20
 - d. Calculate all the vowels in the string
 - e. Remove elements from position 40 to 80
 - f. Insert element "Hi" at index 30
10. Given any 2 strings -
- a. Check if the Strings are equal
 - b. Does both Strings have same prefix (take input strings such that a true condition comes)
 - c. Does both String have same suffix (take input strings such that a true condition comes)

Tuples

11. Create a Tuple of type (String, float), and print each of the String and the float value
12. Create a Tuple of type (Int, Int) representing the point on a graph. Check whether the point
 - a. Is the origin
 - b. Is lying on y-axis (hint : ignore the x type of tuple)
 - c. Is lying on x-axis (hint : ignore the y type of tuple)
 - d. Is in which quadrant of graph (hint : depending on +ve and -ve values of x and y)

Please create a tuple which follows each of these above cases (demo with every type of input)

Dictionaries

13. Create a dictionary of 10 students where the key is roll no. of type Int and the value is name of type String. Perform the following operations on the dictionary -
 - a. Change the name of the student at roll no. 5
 - b. Get the name of the student at roll no. 8
 - c. Remove the student at roll no. 6
 - d. Calculate the total number of students
 - e. Make the dictionary empty
14. Create a dictionary of 5 cities (include an entry 560076 : "Bangalore") and their pin codes where the key is pin code and value is City name. Perform the following operations on the dictionary
 - a. Obtain the city name for pin code - 560076
 - b. Change the pin code for Bangalore (see if it is possible)
 - c. Remove any one city from your dictionary
 - d. Display all the elements in the dictionary
 - e. Make the dictionary empty.
15. Create a bank account dictionary with following details - The key is account number, value is the balance
 - a. Sort the elements of the dictionary in increasing order of balance
 - b. Remove the elements whose balance is less than 100
 - c. Display the element with highest and lowest balance
 - d. Calculate the average balance of all the accounts
 - e. Display all the accounts and their balances
16. Create a dictionary where key is a string and value is a tuple of type Int, Int. Perform these operations on the dictionary
 - a. Sort the elements of the dictionary.

Suggested Teaching Guidelines for

iOS Programming – PG-DMC August 2019

- b. Remove the element whose coordinates are farthest from origin (0,0).
- c. Find the average of the x and y elements (index 1 and 2 of tuple)
- d. Display all the elements of the dictionary.

Sets

17. Create a set of different movie star names. Perform these operations on the set
 - a. Try to add same name in the set. Is it possible ?
 - b. Display all the elements of the set
 - c. Remove the element at position 3
 - d. Check if a given name exist in the set
 - e. Display the total number of names in the set
18. Create a set of 10 numbers and perform these operations on the set
 - a. Sort the elements in ascending and descending order
 - b. Find the average of the elements
 - c. Identify the highest and the smallest element
 - d. Identify if any element is repeated, then how many times (make this true)
 - e. Remove the smallest element in the set
 - f. Print the contents of the set
19. Create a set of tuples of type Int, Int. Perform these operations on the set-
 - a. Sort the elements in ascending order
 - b. Find the average of elements of index 0
 - c. Find highest element of each index
 - d. Print the set.

Assignment for Session 3 – Swift Basics

Note: Write the solution of these problems using Swift language

Functions

1. Write a function to search an element in an array using binary search.
2. Write a function to sort the elements of an array using Quick sort method
3. Write a function that receives a character string and a character as argument and Deletes all occurrences of this character in the string. The function should return the corrected string with no holes.
4. Write a function that would take an Int array as input, and return a Tuple of type (Int, Int). The tuple must have the sum of all the elements of the array and the result of multiplication of all the elements of the array.
5. Write a function to perform matrix multiplication of 2 matrices. The function must return a matrix. Print and display the matrix

Classes and Struct

6. Define a structure called **cricket** which has following info – Player Name, Team Name, Batting average. Use initializer to initiate cricket structure. Using cricket, declare an array players with 10 elements and display the names of the players in order of their batting average.

Suggested Teaching Guidelines for

iOS Programming – PG-DMC August 2019

7. Create a student class which has the following info – Student name, Student roll no, Student branch, Student marks in modules (this must be an array), is student average more than 60% (more than 60% in degree and +2). Use initializer to initiate the Student class.
 - a) Create an array of 10 students.
 - b) Create one more class StudentsOperation. Create a function in this class which will display the list of students in decreasing order of total of their marks in modules. The function takes argument of array of Student objects.
 - c) Create a function in StudentsOperation class Display the list of students who have more than 60%. The function takes argument of array of Student objects.
8. Create a class Car with following info – Car model, Car Type, Car drive type, current km traveled also have functionality of gearTransmission (auto or manual), ignitionStyle(key,keyless), getInfo method to get info of the car. Use initializers to initiate the class.
 - a. Create a class Mercedes which inherits class Car. Override the method gearTransmission (automatic in this case) and method getInfo in it. And display the information about the Mercedes car and its gear transmission.
 - b. Overload the method calculateMileage – one function must take fuel, kms as argument, other function must take fuel amount as arguments.

Assignment for session 4-5 – Swift Basics and Data Structures

Note: Write the solution of these problems using Swift language

Properties

1. Create a class **BankAccount** which has the following attributes – a/c no, name, bank name, amount, duration of deposit.
 - a. Create the a/c number, bank name, a/c holder name as Stored properties
 - b. Create a Computed property interest with the help of amount and time
 - c. Add a property observer for the property – amount and duration of deposit
2. Create a class **IncomeTax** which has the following attributes – Income. Duration, Savings, Tax_calculated. You can also create methods in the class that can be used to compute the property value.
 - a. Store the savings as lazy stored property
 - b. Add property observers for Income and Duration
 - c. Tax calculated is a computed property

For calculating the tax, follow these rules

- a. If income is < 250000, then no income tax
- b. If income is 250000 < 500000, reduce 250000 + savings (which cannot be more than 150000) from the income and calculate the tax as 20% of the remaining amount.
- c. If income is > 500000, calculate tax based on rule a and rule b, and then calculate additional tax of 30% on the remaining amount.

Protocols and Extensions

3. Write an extension of the Double type to perform square root of the value using the function `square_root`

Suggested Teaching Guidelines for

iOS Programming – PG-DMC August 2019

4. Write an extension of the Array to perform sorting of the array using the function sort
5. Create a class **Bird** with properties – name, habitat, food, feather colour. Create a protocol FlyingStyle, which has a method fly (). Conform the protocol to Bird class and write code for the fly method.
6. Create a class **Artist** with properties – name, art_work, expert_level. Create a protocol Creates which has a methods – draw (), sing (), dance (), paint () etc. Implement the protocol methods as per the artist type.

Generics and Data Structures and algorithms

7. Write the program for Selection sort using generics.
8. Write a program for Dequeue using generics.
9. Write a simple generic function to sort an array.

Suggested Teaching Guidelines for
Android Programming – PG-DMC August 2019

Duration: 68 hrs. Class room + 68 hrs. Lab (**136 hrs.**)

Objective: How to effectively build an application on OS used in mobile phones and handheld devices

Prerequisites: Familiar with programming concepts and operating system basics. Students should be proficient in object-oriented programming using java before Symbian OS Module

Evaluation method:	Theory Exam:	40% weightage.
	Practical Exam:	40% weightage.
	Lab performance:	20% weightage.

List of Books / other training material

Courseware:

Beginning Android Programming with Android Studio by J. F. DiMarzio/ Wrox

Reference:

Professional Android Application Development by Reto Meier / Wiley

Language of teaching – Java, Kotlin

Android Studio – v3.0 and above

Note: Each Theory session having 2 Hours and Lab session having 4 Hours.

Session 1,2:

- Introduction to Android Operating System
 - History
 - Various versions of Android.
 - Why develop for Android?
 - Android licencing
- Setting up Android application development environment.
 - Downloading JDK and Android studio.
 - Installing and configuring Android Studio.
- The Developer workflow
- Hello World! - Android
- Understanding an Android Studio Project.
 - Various folders in Android studio project
 - Intro to AndroidManifest.xml
 - Intro to res folder
- What is Gradle?
 - Various aspects of build.gradle file.

Session 3,4,5:

- Introduction to Kotlin Programming Language
 - Advantages of Kotlin over Java
 - Basic Kotlin Syntax

Suggested Teaching Guidelines for
Android Programming – PG-DMC August 2019

- Coding Conventions
- Basic Types, Packages, Control Flow
- Java Interop (Calling java code from Kotlin and vice versa)
- Extension functions
- Co-Routines
- Null-safely, Smart casts, Properties
- First-class delegation, Singletons
- Companion objects
- Data classes

Session 6,7

- Understanding the Android Architecture
 - Android layered Architecture
 - The Kernel Layer
 - The Android Runtime, and Libraries.
 - The Application Framework
- Power Management in Android device.
 - Wake locks
- Intro to IPC in Android
 - IPC using Binder interface
- Dalvik VM vs ART
 - What is DVM?
 - The .dex format
 - ART - Ahead Of Time compilation.
 - Difference between ART and DVM.
- Understanding the Android build system.
 - aapt tool
 - Dex compiler
 - apk builder, Jar signer
 - ZipAlign tool
- Intro to fundamental components of Android OS.
 - Activity
 - Service
 - Content Provider
 - Broadcast Receiver

Session 8,9

- Activity
 - What is an Activity?
 - Activity Life Cycle
 - Various states and lifetimes of an Activity
 - Starting an Activity
 - Starting an Activity for result.
- App Compat library
- Intent
 - What is an Intent?
 - Various uses of Intent.

Suggested Teaching Guidelines for
Android Programming – PG-DMC August 2019

- Intent: An IPC mechanism
- Types of Intent
 - Implicit intent
 - Explicit intent
- Intent Filter
 - Intent filter options - action, data, category
- Context
 - What is Context?
 - Various uses of context.
- Data sharing using Intent
 - Sharing primitive data using Intents
 - Sharing objects using Intents
- Navigation and Task Stack

Session 10,11

- Basic Android UI
 - What is a View, View Group and Layout?
 - Android View System
- Android Layouts
 - Linear layout
 - Relative layout
 - Frame layout
 - Constrained layout
- Various View attributes
- Button, TextView and EditText
- Various UI Events and Event listeners.
- Rendering web content through Webview
- ScrollView

-Session 12,13

- Spinner
- Checkbox
- Radio Button
- Date and Time Pickers
- Switch and Toggle
- ImageView
 - Supporting various screen sizes
- Drawable
 - Shape Drawable
 - State list Drawable
 - Layer list Drawable
 - Nine patch Drawable
- Styles and Themes
 - Default themes

Suggested Teaching Guidelines for
Android Programming – PG-DMC August 2019

- Creating custom theme
- Inheriting themes.

Session 14,15

- Animations
 - Property Animation
 - Tween Animation
 - Frame Animation
- Notifications
 - Basic
 - Expanded
- Progress dialog
- Introduction to app Widgets
- Media Player
 - Playing Audio
 - Playing Video

Session 16,17

- Fragment
 - What is a Fragment?
 - Creating a Fragment
 - Fragment Life Cycle
 - Communication between Fragment and Activity
 - Fragment Transactions: Add, Replace, and Remove fragments
- Master Detail View using Fragments
- Types of Fragments
 - Dialog Fragment
 - Preference Fragment

Session 18,19

- ListView: Displaying lists in Android
 - What is an Adapter?
 - Simple lists using ArrayAdapter.
 - Custom List using BaseAdapter
 - Faster Scrolling through ViewHolder.
 - Various events on ListView items.
- Introduction to Material Design
 - What is Material Design?
 - Adding material design support to your app.
 - Material design colour scheme and Themes
- Introduction to RecyclerView
 - ListView vs RecyclerView: Advantages of RecyclerView
 - Implementing RecyclerView.Adapter class
 - Implementing RecyclerView. ViewHolder class.

Suggested Teaching Guidelines for
Android Programming – PG-DMC August 2019

- Handling click events in RecyclerView
- Floating Action Button
- SnackBar

Session 20,21

- ActionBar
 - Adding ActionBar items
 - Handling click events on ActionBar items.
- ToolBar
 - Using ToolBar as ActionBar
- Menus and Pop ups
 - Contextual Action Modes
 - Pop up Menu
 - Pop up Window
- Coordinator Layout
- Tab based UI using View Pagers
- Navigation Drawer
- Collapsing ToolBar

Session 22,23

- Android Storage
 - Storing data in Shared Preferences
 - Storing and retrieving data into internal memory and cache.
 - Storing and retrieving data into external memory and cache.
 - Storing and retrieving objects from internal and external memory.
 - Accessing /res and /res/raw folder files.
- Storing organized Data into SQLite DB
 - Content Values and Cursors.
 - CRUD operations on SQLite DB
 - Searching from table in SQLite DB.

Session 24,25

- Content providers
 - Access mobile contacts/SMS using Content Provider.
 - Create content provider for custom DB.
- Multi Tasking in Android
 - Thread mechanism in Android
 - Java Threads and Runnable interface
 - Threads and Thread Handlers.
 - AysncTasks

Session 26,27

Suggested Teaching Guidelines for
Android Programming – PG-DMC August 2019

- Services in Android
 - What is a Service?
 - Types of Service - Started Service, Bounded Service
 - Service Life cycle
 - Starting a service
- IPC between Service and Activity using Binder
- Intent Service
- Broadcast Receiver
 - Handle custom app based event.
 - Handle system event.

Session 28,29

- Android Location Based Services
 - Fused Location using Google API
 - Displaying Map, marking the current location, Adding various map options.
 - Regular location update using LocationListener
 - Geocoding
- Android Sensors
 - Obtain values from motion and direction sensors - Compass, Gyroscope, Accelerometer
 - Making use of proximity, light sensor.
- Managing Wi-Fi programmatically.
- Bluetooth communication.
- Telephony Manager - Call and SMS.

Session 30,31

- Consuming REST API
 - Through HttpURLConnection/HttpsURLConnection class
 - Using Retrofit 3rd party library.
- JSON Parsing
- Using Picasso library for managing image downloading.
- Using Social media
 - Facebook Login via Facebook SDK
 - Google Login via Firebase
 - Twitter Login via Firebase
- Simple User management using Firebase

Session 32,33

- Push Notifications
 - Push Notifications from BaaS (Firebase, Parse etc).
 - Sending push notifications from custom server
- Intro to Git/GitHub VCS from Android Studio
- Intro to Android Design Patterns.

Suggested Teaching Guidelines for

Android Programming – PG-DMC August 2019

- MVP Design pattern
- Intro to app uploading process on Google Play.

Session 34

- What is Reactive Programming
- Introduction to RxJava
- Sample usage
- Integrating popular libraries (e.g. Agera) for writing reactive applications.

Assignments Session 1 – Android

1. Setup Android app development environment.
2. Create a Hello World project. And show the output in Emulator.

Assignments Session 2, 3 – Android

3. Create a basic calculator app with 4 operations – Add, Sub, Divide, Multiply.
4. Create a basic temperature conversion app – Provide an input field to insert temperature to convert. Provide 2 buttons – one for F to C conversion, other for C to F conversion. And 1 Text View to show the output temperature.
5. Create an app to check whether a number is prime/palindrome or not.
6. Create an app which will read the contents of text entered in a text field and display the number of words in the field. Also, print the text written in the text field in a text view.
7. Create an Android app which has a student registration form with fields – name, roll no, branch, course name. Check whether the user has entered all the fields or not. If the user has entered all the fields. Click on a submit button. After click on submit button, open 2nd activity, where all the information is displayed in a text view.
8. Create an Android application that will ask user to open the settings page. Check whether the settings activity has opened successfully.
9. Create an android application which will dial the number written in a text field. Also, create another text field, in which you have to write a web address, which will be opened. Use implicit intent for the same.

Assignments Session 4, 5 – Android

10. Create a basic app to access the location of your device. Design the app such that it works for Android v7.1.1. (New Android Permissions)
11. Create airline registration form which has following fields – Name (Edit Text), Age(number), Gender(radius button), Travelling to (Spinner), Travelling From (Spinner), No. of seats (number edit text), Food preference (veg and non veg with radio buttons), Preferred time of flight (Session, night, afternoon – with check boxes, can select more than 1 option), Credit card number (Number with only 16 digits allowed), CVV(password with only 3 digits allowed), and Buttons Submit and Exit.
 - a) Check if the user has filled all the options correctly after click on submit button. If a field is not properly filled, change the background colour of that field to red, and others to green.

Suggested Teaching Guidelines for

Android Programming – PG-DMC August 2019

- b) Display all the details with their labels in a new activity after all the details are filled correctly and submit button clicked. Exit the application after the exit button is clicked.
- c) Prepare the UI for this form in both linear layout and Relative layout.
- d) Use scroll view if all the elements are not fitting in the screen.
- 12. Create an employee registration form which has following fields – Name (text), Age (number), company name(text), department(spinner with multiple options), designation(text), Pay scale(number), Years of experience(number), Areas of work(multiline text supporting edit text) and buttons Submit and exit.
 - a) Check if the user has filled all the options correctly after click on submit button. If a field is not properly filled, change the background colour of that field to red, and others to green.
 - b) Display all the details with their labels in a new activity after all the details are filled correctly and submit button clicked. Exit the application after the exit button is clicked.
 - c) Prepare the UI for this form in both linear layout and Relative layout.
 - d) Use scroll view if all the elements are not fitting in the screen.
- 13. Create an Android app which does the following –On click of a button, shows the webpage www.google.com in the same activity using Webview.
- 14. Create a theme changer application, which would change the theme of the current application. In the app page, show a spinner, which will have options for various themes. When either of the theme is selected, the theme must be applied on the app.
- 15. Create an app which makes uses of all types of Drawable.

Annexure

Assignments Session 6 – Android

1. Create an application which demonstrates different types of animation
 - a. Animate an ImageView using property animation.
 - b. Create a frame animation using multiple images.
 - c. Demonstrate tween animation.
2. Create an application which contains a basic registration form with details – Name, email, username, password and a Submit button. When the user clicks on Submit button, an expanded notification must be generated and shown. When the notification is expanded, it must display the Name and user name of the user.
3. Create a simple media player application, which plays an audio song, play a video from /res/raw folder, also try to play song from phone memory.
4. Create an application which creates a small widget.

Assignments Session 7 – Android

1. Create a basic username-login demo app, such that, the login page is loaded through the LoginFragment. Once the user log in successfully, the welcome screen should be displayed from WelcomeFragment. Replace the LoginFragment with WelcomeFragment.
2. Create an application which demonstrates the master detail flow UI pattern.
3. Create an application to display a simple dialog using dialog fragment.

Assignments Session 8 – Android

Suggested Teaching Guidelines for

Android Programming – PG-DMC August 2019

4. Create the To-do application using a Custom Adapter. Provide the functionality to delete an entry on long press on any list entry. Also, the list must be updated automatically if there is any change in data.
5. Create an application which has a form for Student registration, which has following fields – Name (text), Age (number), Branch(text), Course(text), Average marks(Decimal number), Grade(text), Technical strength(text) and button Submit.
 - a. When the submit button is clicked, the information must be stored into a Student object instance which has these same class variables. This student object must be stored in an ArrayList of Student objects.
 - b. Enter details of at least 5 students.
 - c. Display the details (mentioned above) of each student in a List using List View. (Hint: Use Custom adapter for managing the list).
6. Create a mini m-commerce application. When the app opens for the first time, display the list of available products. Each entry of the product display list must have the following entries – Product Image (Image View), Product Name(Text), Product Description(text), Add to Cart(Button), Details(Button). Also provide an Action item of cart in the same activity, to access the cart at any time.
 - a. When the user clicks on the buy button, add the product into the cart.
 - b. When the user clicks on the cart action item, open an activity, which has the list of all the items in the cart. Provide a button – remove from cart (which when clicked will remove the product from the cart) in each cart entry. After the list, put a button – Proceed to Checkout (Button). When this button is clicked, display a toast, which has info about total number of items and the total cost of the cart.

Assignments Session 9

1. Create the home page for DMC's android app
 - a. The home page must contain a navigation drawer with options – About, Modules, Faculties, Info, Gallery
 - i. About – A new activity which has a TextField which has text about DMC
 - ii. Modules – A new activity which has list of all the modules in DMC
 - iii. Faculties – List of faculties and their short description. Create a list with custom row having name, degree, designation, image, description. This list must be in an activity which has collapsible toolbar
 - iv. Info – Other imp info
 - v. Gallery – Create a gallery using Tab based activity, where each page has 3 image views with images. Use any image you want.
 - b. The Home page UI must contain 3 Fragments in 3 FrameLayouts. This UI must be in a nested scroll view.
 - i. Fragment 1 – A Big TextView displaying total current strength of class
 - ii. Fragment 2 – A Big TextView displaying placement percentage of class
 - iii. Fragment 3 – A list of companies which had come for on campus placement.
2. Create a city tourist app. The home page of the app must display a list of all tourist destination in your city. When any place is clicked, a new activity must be opened. In the new activity, using

Suggested Teaching Guidelines for

Android Programming – PG-DMC August 2019

Coordinator layout and collapsible toolbar layout, create a details page such that the toolbar must contain an image of the place, and the description of the place must be below it.

Assignments Session 10

1. Create a Notes app, which stores the notes taken down by the user in a file in the phone's external memory. When the app loads, it should fetch the last written note in the file.
2. Create an application which has a form for Student registration, which has following fields – Name (text), Age (number), Branch(text), Course(text), Average marks(Decimal number), Grade(text), Technical strength(text) and button Submit.
 - a. When the submit button is clicked, the information must be stored into a Student object instance which has these same class variables. This student object must be stored in SQLite DB.
 - b. Enter details of at least 5 students.
 - c. Display the names of students (mentioned above) in a List using List View when clicked on an action item show in the main activity. Display the list view in the same activity. Use a fragment to display list view
 - d. Display the details of a student in a dialog when any list item is clicked.
 - e. Delete the entry of the student when there is a long press on any student entry.

Assignments Session 11

1. Create an application which calculates the compound interest for input of – Principle, Rate, duration, where rate is compounded annually. Use Async Task to perform calculation in background and display the result in a label.
2. Create an application which reads a file from the phone memory and displays the content in a label. Use AysncTasks for reading the file in background.
3. Create an application to display all the contacts present on the device. Make use of Content providers for the same.

Assignments Session 12

1. Create an application, which uses a service to download an image from the web. And display the image on an ImageView.
2. Create an app which makes use of an Intent Service.
3. Create an app such that, it constantly checks if internet is on. If the state of internet changes, notify the user about is using a snack bar.

Assignments Session 13

1. Create a location manager application. On the home page, the application must display the current location on the Google map, with a marker. Attach a location listener to monitor location changes. On the click of menu item address, open a new activity, which displays the nearby addresses using Geocoding capability.
2. Create an application which shows the value of the sensors – Accelerometer, gyroscope, and Compass. Display the direction of device in degrees.
3. Create an application to display all the nearby Wi-Fi hotspots in a list. Connect to the Wi-Fi hotspot selected.

Suggested Teaching Guidelines for
Android Programming – PG-DMC August 2019

Assignments Session 14

1. Create an application which displays the current weather of your city. Use location-based services for detecting the location of your city. Obtain the details of the weather from the weather.com website and use their REST API to get the latest info. Use Retrofit library to manage the REST calls. Use Picasso library for managing the image downloading.
2. Create an application to display all the contacts present on the device. Make use of Content providers for the same.

Assignments Session 15

Create a push notification application. Which receives push notifications from a web interface designed using Angular JS, Bootstrap running on Node JS server.

- Case Study
 - Augmentated reality of application
 - Integration with social networking website

Assignments Session 16 & 17

Create any eight above Android programs using Kotlin Language (Follow MVP Design patterns to implement applications).

Assignments Session 18 & 19

Implement concurrent ,functional reactive applications using RxJava

Suggested Teaching Guidelines for

Hybrid Mobile Apps Programming - PG-DMC August 2019

Duration: 40 class room hours + 40 lab hours **(80 hrs.)**

Objective: To reinforce knowledge of Hybrid Mobile Apps Programming

Evaluation method:

Theory exam–	40% weightage
Lab exam –	40% weightage
Assignments –	20% weightage

List of books / other training Material:

Mobile App Development with IONIC 2 by Chris Griffin / O'reilly

Reference:

1. HTML5 covers CSS3 , Java script, XML , XHTML, Ajax, PHP and jQuery- Black Book/Dreamtech Press
2. Internet & World Wide Web How to Program by Deitel, Deitel & Nieto; Pearson Education
3. XML - How to Program - Deitel and Deitel - Prentice Hall
4. Ajax In Action by Dave Crane, Eric Pascarello, Dreamtech Press
5. Web Application Security: A Beginner's Guide, Bryan Sullivan, TMH
6. Joomla: 24-Hour Trainer (With CD)by Jen Kramer/ Wiley India Pvt Ltd (2011)
7. Node.js, MongoDB and AngularJS Web Development by Dayley Brad Dayley / Pearson
8. Pro AngularJS by Klaus Freeman Adam Freeman Freeman / Apress
9. AngularJS Services by Paperback / Shroff Publishers & Distributors
10. Angularjs in Action by Lukas Ruebbelke / Dreamtech Press
11. JavaScript: The Good Parts 1st Edition by Douglas Crockford / O'Reilly
12. JavaScript Step by Step 2nd Edition by Steve Suehring/PHI Learning
13. HTML5, JavaScript and jQuery by Dane Cameron/Wiley India
14. Sams Teach Yourself JavaScript in 24 Hours (English) 1st Edition Phil Ballard , Michael Moncur/Pearson Education
15. Moncur/Pearson Education
16. High Performance JavaScript 1st Edition by ZAKAS / O'Reilly

Note: Each session having 2 Hours

Session 1 & 2: HTML

- Introduction to HTML5, Basic HTML, Introduction to basic HTML Aligning the Headings
- Anchor Tag, Paragraph, Images and Pictures, Tables, Framesets
- New features in HTML5, New element, New attribute, Link relations,
- Multimedia, HTML Forms, HTML Controls, INPUT
- Text Area, Radio Button, Check Box, Dropdown, List box, Submit button, Set button
- Button, List and Tables
- HTML 5, Elements, Objects, Events, Canvas, HTML5 Validation, Audio & Video Support

Assignment – Lab:

- Create a webpage with HTML describing your department. Use paragraph and list tags. Apply various colors to suitably distinguish key words. Also apply font styling like italics, underline and two other fonts to words you find appropriate. Also use header tags. Create links on the words

Suggested Teaching Guidelines for

Hybrid Mobile Apps Programming - PG-DMC August 2019

e.g. “Wi-Fi” and “LAN” to link them to Wikipedia pages. Insert an image and create a link such that clicking on image takes user to other page. Change the background color of the page. At the bottom create a link to take user to the top of the page.

- Create your bio-data in an HTML Page. Divide it into following sections – Personal information, Family Background, Academic Qualifications, and Experience. Now divide a HTML page into three frames as upper, left and right (main) frames. Write a Heading in the upper frame and put the bio-data sections links in the left frame and on click the section links the respective detail information should be displayed into the right main frame.
- Create a HTML Strict document. Use HTML comments, the title element, and meta elements to provide information about a document. Use headings, paragraphs, lists (definition, bullet, and numbered), blockquote, address, and preformatted block elements. Use line breaks, hypertext links, images, strong, emphasis, code, superscript, and subscript inline elements. Use id's and internal anchor links to jump within a page
 - o For Lab, there are two members in a team, one team member (the driver) has control of the keyboard/mouse and actively implements the program
 - o The other team member (the navigator) continuously observes the work of the driver to identify tactical defects (such as syntactic and spelling errors, etc.) and also thinks strategically about the direction of the work. Please check as follows for lab work:
- Create a structured XHTML 1.0 Strict document with html, head, and body elements
- Use HTML comments, title, and meta elements to a web document
- Switch pair programming roles
- Use headings to create a document outline
- Switch pair programming roles
- Format text with block tags
- Format text with inline and anchor tags
- Screen Snapshots

Session 3: Working with CSS

- Introduction to CSS, Styling HTML with CSS, Structuring pages with CSS, Cascading style sheet
- Inline Styling (Inline CSS), External Styling (External CSS), CSS Fonts, The CSS Box Model
- The id Attribute, The class Attribute, HTML Style Tags, Linking a style to an HTML document,
- Internal style sheet, Multiple styles

Assignment Lab:

- Considering the lab assignments of HTML
- Write a CSS rule that changes the color of all the elements with attribute CLASS =”Green-Move” to green and shift them down 25 pixels and right 15 pixels.
- Set the page body's margins to 10 pixels on all sides with no padding. Use a Verdana sans-serif font and set the default font-size to 95%
- Set all divisions to display with a 1 pixel, solid, black border (to help with debugging) In the Header Area
- Set its heading 1's text to display centered, in a 'Courier New' mono space font, and a font size of .8 em
- Set its heading 2's text to display centered with a font size of 1.5 em In the Navigation Area
- Make its width 25% of the screen size and float it left

Suggested Teaching Guidelines for

Hybrid Mobile Apps Programming - PG-DMC August 2019

- Style its unordered list of anchor elements to look like a menu
- Do not add the background image or onfocus attribute (final optional steps) in tutorial
- use a background color of red and a text color of white
- center the text
- In the Main Content Area
- Set the right and left margins to 26% of the screen size
- Make its heading 2's text centered with a font size of 1.2 em
- For Printing, write the style rules to
- Set display to none for the navigation and advertisement areas so that only the header and main content are printed.
- You do not need to specify style rules for the header and main content elements.
- The browser should use the default settings when printing your Web page
- Create a form to submit a resume
- Using in line CSS. Write all the above styling in CSS in different file (.css) and link it to your webpage such that changes made in CSS file are immediately reflected on the page. Group paragraphs into single class and add styling information to the class in CSS. Add few form elements such as radio buttons, check boxes and password field. Add a submit button at last.
- Apply inline CSS to change colors of certain text portion, bold, underline and italics certain words in your HTML web page. Also change background color of each paragraph
- Create a simple form to submit user input like his name, age, address and favourite subject, movie and singer.

Session 4 & 5: Learning JavaScript

- Introduction to JavaScript
- Advantages of using Java Script on client side over VB Script
- Variable, statements, Operators, Comments, constructs, Functions, expressions
- Variables in Java Script
 - o "Var" type
 - o Scope of variables
- JavaScript console
- Scope, Events, Strings, String Methods, Numbers, Number Methods, Dates, Date Formats, Date Methods
- How to handle events?
- Arrays, Array Methods in JavaScript
- Using array methods (length, reverse, sort etc)
- Creating Objects in Java Script
 - o Date
 - o String
 - o Using Object methods
- Operators
 - o Arithmetic
 - o Logical
 - o Bitwise
 - o this

Suggested Teaching Guidelines for

Hybrid Mobile Apps Programming - PG-DMC August 2019

- o new
- o delete
- Booleans, Comparisons
- Control and Looping Statements
- Conditions, Switch,
- Loop For, Loop While, Break
- Functions, Function Definitions, Function Parameters, Function Invocation, Function Closures
- Common Events
 - o onClick
 - o onLoad
 - o onMouseOver
 - o onReset
 - o onSubmit
- Different functions:
 - o alert(), prompt(), confirm().
 - o eval
 - o isFinite
 - o isNaN
 - o parseInt and parseFloat
 - o Number and String
 - o escape and unescape

Assignment – Lab:

- You are having sample html file with a submit button. Now modify the style of the paragraph
- text through javascript code
- Implementing mathematical assignments like factorial, Matrix Multiplication, etc. in JavaScript
- Writing programs to design a form to input student's data for registering a course.
- Writing programs for implementing user inputs and displaying the same. Use HTML to design
- input and output forms.
 - o Write functions to implement the above assignments.
- Write a program to sort input strings.
- Display a complete date with the name of the Session and name of the month

Session 6: Learning JavaScript (contd.)

- Objects, Object Definitions, Object Properties, Object Methods, Object Prototypes
- Object Oriented Programming in JavaScript
 - o Method
 - o Constructor
 - o Inheritance
 - o Encapsulation
 - o Abstraction
 - o Polymorphism

Suggested Teaching Guidelines for

Hybrid Mobile Apps Programming - PG-DMC August 2019

- Javascript Validations

Assignment – Lab:

- Validate the above resume form using the Java Script
- Design Different windows forms to input user commands and process them through output screen
- Write a JavaScript program to get the website URL
- Write a JavaScript program to list the properties of a JavaScript object
- Write a JavaScript function to get the values of First and Last name of the following form
- Write a JavaScript program to set the background color of a paragraph.
- Write a JavaScript function to add rows to a table
- Write a JavaScript program to display the reading status (i.e. display book name, author name and reading status) of the following books
- Write a JavaScript function to validate whether a given value is RegExp or no
- Write a JavaScript function to check whether given value types are same or not.
- Write a JavaScript function to validate whether a given value type is Boolean or not
- Write a JavaScript program to sort a list of elements using Quick sort.
- Write a JavaScript program to sort a list of elements using Insertion sort.
- Write a JavaScript program to sort a list of elements using Bubble sort

Session 7 & 8: Working with jQuery

- Introducing to jQuery
- Basics of jQuery coding
- Selecting the elements
- Bringing pages to life with jQuery
- jquery selection and events
- JQuery Events
- Energizing pages with animations and effects
- jquery Effects
- jquery traversal and manipulation
- Data attributes and templates
- DOM with jQuery utility functions
- jQuery Plugins
- jQuery / Google Web Toolkit

Assignment – Lab:

- Write a jQuery Code to get a single element from a selection, you have the HTML page.
- Write jQuery Code to add a tag at the beginning of the list item, containing the index of the list item
- You are having sample data for the link. Write jQuery Code to change the hyperlink and the text of an existing link.
- Write jQuery for the following:
 - o Attach a click and double-click event to the <p> element.
 - o Hide all headings on a page when they are clicked
 - o Find the position of the mouse pointer relative to the left and top edges of the document

Suggested Teaching Guidelines for

Hybrid Mobile Apps Programming - PG-DMC August 2019

- o Count the number of milliseconds between the two click events on a paragraph.
- o Display the window width while (or after) it is resized
- o Count the number of specific elements
- o Set a timer to delay execution of subsequent items in the queue.
- o Create a Zebra Stripes table effect
- o Print a page using jQuery
- o Make first word bold of all elements

Session 9 & 10: Working with Node.js

- Introduction to Node.js
- Node modules
- Selectors
- Syntax
- Developing node.js web application
- Event-driven I/O server-side JavaScript

Assignment – Lab:

- Installing node.js
- Working with Node modules
- Developing node.js web application
- Write a recursive function in Node.js
- Write a Node program that prints all the numbers between 1 and 100, each on a separate line. A few caveats:
 - o if the number is divisible by 3, print “fizz”
 - o if the number is divisible by 5, print “buzz”
 - o if the number is divisible by both 3 and 5, print “fizzbuzz”
- Use Node and Express to write a simple web application that checks whether a number is a prime or not and display message accordingly.

Session 11 & 12: Working with Express

- Introduction to Express
- First Express Application
- Application, Request and Response Objects
- Implementing MVC Pattern
- Express application configuration
- Rendering Views

Assignment – Lab:

- Installing Express
- Build any previous assignment using Node.js web application using Express Application
- Create a web application using Node and Express framework

Session 13 & 14: Working with AngularJS

- Introduction to AngularJS
- Structuring AngularJS application
- MVC in AngularJS

Suggested Teaching Guidelines for

Hybrid Mobile Apps Programming - PG-DMC August 2019

- Anatomy of an AngularJS App
- Using Binding and Template Directives
- Using Element and Event Directives
- Working with Forms
- Using Controllers and Scopes
- Working with Modules and Services
- Services for Global Objects, Errors and Expressions
- AngularJS routing
- AngularJS services
- Using Filters
- Creating Custom Directives
- Creating Complex Directives
- Advanced Custom Directive Features
- Services for REST
- Services for Views
- Services for Animation and Touch
- Services for Provision and Injection
- Services for Testing"

Assignment – Lab:

- Installing AngularJS
- Writing simple application on Angularjs like to design forms, data inputs etc
- Writing programs to design a form to input student's data for registering a course.
- Write a program to implement REST services.
- Validate the above programs forms using the Angular Java Script

Session 15,16,17: Mobile Back End As a Service

- Mobile Backend as a Service -characteristics, services
- Loopback.io framework - REST API Design, Push Notifications
- Mobile Client SDKs - Android, iOS

Session 18: Responsive Web Design

- Introduction of UI Scripting
- The Best Experience for All Users
 - o Desktop
 - o Tablet
 - o Mobile
- Bootstrap
- Overview of Bootstrap
 - o Need to use Bootstrap
 - o Bootstrap Grid System, Grid Classes, Basic Structure of a Bootstrap Grid
 - o Typography
 - o Tables, Images, Jumbotron, Wells, Alerts, Buttons, Button Groups, Badges/Labels, Progress Bars, Pagination, List Groups, Panels, Dropdowns, Collapse, Tabs/Pills, Navbar, Forms, Inputs
 - o Bootstrap Grids, Grid System, Stacked/Horizontal
 - o Bootstrap Themes, Templates

Suggested Teaching Guidelines for

Hybrid Mobile Apps Programming - PG-DMC August 2019

Assignment Lab:

- Design of a Login and Registration pages using above Bootstrap tools
- Testing of web pages designed in HTML5 and modifying the UI as per the above features.

Session 19 & 20: Flutter

Lecture

Introduction to Flutter

- User Interface building
- Data & backend
- Platform integration
- Packages & plugins

Flutter Deployment

- Build and release for Android
- Build and release for iOS
- Continuous deployment with fastlane

Assignment Lab:

- Design a simple calculator using Flutter and deploy in both android & iOS platform.
- Design an app that includes login/signup, listView/gridview, google map and or firebase.

Suggested Teaching Guidelines for
Aptitude & General English – PG- DMC August 2019

Duration: 50 class room hours

Objective: To reinforce knowledge of general Aptitude & English

Prerequisites: Knowledge of Mathematics & English.

Evaluation method: Theory exam– 100% weightage

List of Books / Other training material

Reference:

1. Quicker math by M. Tyra (BSC publication co. Pvt. Ltd)
2. Quantitative Aptitude by RS Aggarwal
3. Verbal & Non-Verbal Reasoning: RS Aggarwal
4. Quantitative Aptitude - Quantum CAT : Sarvesh K Verma
5. High School English Grammar & Composition Revised Edition Wren, Martin / S. Chand Publisher
6. How to prepare GRE by Barron's / galgotia publications pvt. Ltd
7. Oxford Guide to English Grammar 01 Edition John Eastwood / Oxford University Press

Website to refer: www.indiabix.com

Note: Each session having 2 Hours

Part I – Aptitude

Session 1:

Lecture

- ° Analogy
- ° Series Completion (Number, Alphabet, Letter Series)
- ° Coding-Decoding for Number, Alphabet and Letter

Session 2:

Lecture

- ° Blood Relations
- ° Puzzle Test
 - Classification Type questions
 - Compression Type questions
 - Sequential order questions
 - Section based on given conditions
 - Questions involving family members

Session 3:

Lecture

- ° Alphabet test
- ° Order of words

Suggested Teaching Guidelines for
Aptitude & General English – PG- DMC August 2019

- ° Letter words problems
 - Rule detection
 - Alphabetical quibble
 - Word formation
- ° Logical sequence of words

Session 4:

Lecture

- ° Number, Ranking and time Sequence Test
- ° Mathematical operations
- ° Arithmetic reasoning

Session 5:

Lecture

- ° Logical reasoning
- ° Statement-Arguments
- ° Statement-Assumptions

Session 6:

- ° Statement-courses of Action
- ° Statement-Conclusions
- ° Deriving conclusion from passages

Session 7:

- ° HCF and LCM
- ° Fraction
- ° Number system
- ° Permutation & combination

Session 8:

- ° Ratio & Preparation
- ° Partnership

Session 9:

Lecture

- ° Average
- ° Percentage

Session 10:

Lecture

- ° Clock
- ° Probability

Session 11:

Lecture

- ° Pipes and cisterns
- ° Problem on streams

Session 12:

Lecture

- ° Time and work
- ° Work and Wages

Session 13:

Lecture

Suggested Teaching Guidelines for
Aptitude & General English – PG- DMC August 2019

- ° Problem on Trains
- ° Problem on Speed and Velocity

Session 14:

Lecture

- ° Problem on Ages
- ° Profit and loss

Session 15:

Lecture

- ° Simple Interest,
- ° Compound Interest

Part II – General English

Session 16:

Lecture

- ° The Sentence
 - Subject and Predicate
 - Phrase and Clause
 - Parts of Speech
- ° The Noun: Kinds of Nouns
 - The Noun: Gender
 - The Noun: Number
 - The Noun: Case

Session 17:

Lecture

- ° The Adjective
 - Comparison of Adjectives
 - Adjectives Used As Nouns
 - Position Of The Adjectives
 - The Correct Use Of Some Adjectives
- ° Articles
 - Personal Pronouns
 - Reflexive And Emphatic Pronouns
 - Demonstrative, Indefinite And Distributive Pronouns
 - Relative Pronouns
 - Interrogative Pronouns

Session 18:

Lecture

- ° The Verb
 - Verbs Of Incomplete Predication
 - Active And Passive Voice
- ° Mood
 - Tenses: Introduction
 - The Uses of The Present And Past Tenses
 - The Future
 - The Verb: Person And Number

Suggested Teaching Guidelines for
Aptitude & General English – PG- DMC August 2019

- The Infinitive
- The Participle
- The Gerund
- Irregular Verbs
- Auxiliaries And Modals
- Conjugation of The Verb Love

Session 19:

Lecture

- ° The Adverb
- ° Comparison Of Adverbs
- ° Formation Of Adverbs
- ° Position Of Adverbs
- ° The Preposition
- ° Words Followed By Prepositions
- ° The Conjunction
- ° Some Conjunctions And Their Uses
- ° The Interjection
- ° The Same Word Used As Different Parts Of Speech

Session 20:

Lecture

- ° Composition
- ° Analysis, Transformation and Synthesis
- ° Analysis of Simple Sentences
- ° Phrases
- ° Clauses
- ° Sentences: Simple, Compound and Complex
- ° More about Noun Clauses
- ° More about Adjective Clauses
- ° More about Adverb Clauses

Session 21:

Lecture

- ° Analysis of Complex Sentences
- ° Analysis of Compound Sentences
- ° Transformation of Sentences
- ° Transformation of Sentences
- ° Synthesis of Sentences
- ° Synthesis of Sentences
- ° Synthesis of Sentences
- ° The Sequence of Tenses
- ° Direct and Indirect Speech

Session 22:

Lecture

- ° Agreement of The Verb With The Subject
- ° Nouns and Pronouns
- ° Adjectives

Suggested Teaching Guidelines for
Aptitude & General English – PG- DMC August 2019

- ° Verbs
- ° Adverbs
- ° Preposition
- ° Conjunctions

Session 23:

Lecture

- ° Order of Words
- ° Synonyms & Antonyms
- ° Punctuation
- ° Spelling Rules
- ° The Formation of Words
- ° Figures of Speech Exercise

Session 24:

Lecture:

- ° Verb Patterns
- ° Question Tags, Short Answers, Etc.
- ° More Structures

Session 25:

Lecture

- ° Words of Idioms & phrases
- ° Sentence Construction
- ° Fill up the blanks

Suggested Teaching Guidelines for **Effective Communication – PG-DMC August 2019**

Duration: 50 class room hours

Objective: To reinforce knowledge of communication

Prerequisites: Knowledge of English.

Evaluation method: Theory – 100% weightage

List of Books / Other training material

Reference:

1. Business Communication by H S Mukerjee / Oxford University Press
2. Business Communication by R K Madhukar / Vikas Publishing House Pvt. Ltd.
3. Business Communication Essentials A skills-Based Approach to Vital Business English by Courtland Bovee, John Thill / Pearson
4. Effective Business Communication by Asha Kaul / Prentice Hall of India
5. Fundamental of Technical Communications by Meenakshi Raman, Sangeeta Sharma / Oxford University Press
6. English is easy by Chetan Anand Singh/ BSC publication Co. Pvt. Ltd
7. Communication Skills Publication Year 2011 Sanjay Kumar, Pushp Lata / Oxford University Press
8. Professional Communication Skills Praveen S R Bhatia / S. Chand Publishing

Note: Each session having 2 Hours

Session 1:

Lecture

Discussion on Soft Skills
Intro to communication
Effective Communication

Session 2:

Lecture

Communication Skills and its practice
Barriers to communication
Kind of communication

Session 3:

Lecture

Pre assessment Test
Official & General Conversation
Confidence building

Session 4:

Lecture

Work place Etiquettes
Official Letter Writing
Confidence building

Session 5:

Lecture

Suggested Teaching Guidelines for
Effective Communication – PG-DMC August 2019

Official Emailing
Event Reporting
Confidence building

Session 6:

Lecture

Non-verbal Communication
Gestures

Session 7:

Lecture:

Fluency and vocabulary
Essay Writing

Session 8:

Lecture:

Pronunciation
Conversation practice
Polite Conversation

Session 9:

Lecture:

Greeting
Formal Etiquettes
Body Language

Session 10:

Lecture

Writing Covering letter
Email

Session 11:

Lecture

Writing Resume

Session 12:

Lecture

Post Assessment Test
Personal Goal Setting & Career Planning

Session 13:

Lecture

Presentation Skill

Session 14:

Lecture

Formal Speaking (Telephone, Face-to-Face, Public Speaking)
Oral & Digital Presentation Skills

Session 15:

Lecture

Presentation Skill
Developing Positive Attitude

Session 16:

Lecture

What is group discussion?
Group Discussions

Suggested Teaching Guidelines for
Effective Communication – PG-DMC August 2019

Session 17:

Lecture

Group discussion on at least 4 topics

Session 18:

Lecture

Group discussion on at least 4 topics

Session 19:

Lecture

Interview skills

Job Search Process

Session 20:

Lecture

Interview skills

Listening Skills

Session 21:

Lecture

Interview skills

Audio Synthesis

Session 22:

Lecture

Practice of Mock interview

Session 23:

Lecture

Practice of Mock interview

Session 24:

Lecture

Practice of Mock campus

Session 25:

Lecture

Mock campus

Cross-Cultural Communication

Technology-enabled Communication