BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, Pilani
Pilani Campus
Instruction Division

## SECOND SEMESTER 2013-2014

## Course Handout Part II

In addition to part-I (General Handout for all courses appended to the time table) this portion gives further specific details regarding the course.

**Course No.** : CS F111
**Course Title**: Computer Programming

**Instructor-in-Charge**: VISHAL GUPTA (email: vishalgupta@pilani.bits-pilani.ac.in)

**Instructors:**

| **Name** | **EMail@pilani.bits-pilani.ac.in** | **Name** | **EMail@pilani.bits-pilani.ac.in** |
|---|---|---|---|
| Sundaresan Raman | sundaresan.raman | Chandramani Chaudhari | chandramani.chaudhary |
| Asma Rani | asma.rani | Sonal Kumari | sonal.kumari1910@gmail.com |
| Avinash Gautam | avinash | Jagat Sesh Challa | jagatsesh |
| K Manjusha | manjusha | Saiyedual Islam | sislam |
| Mayuri Digalwar | mayuri | Mohit Sati | mohitsati1@gmail.com |
| Murali P | muralip | Rupal Bhargava | rupal.bhargava |
| N Mehala | mehala | Neetika Gupta | neetika.gupta |
| Shailendra S Shekhawat | shailu | Rajkiran Reddy | rajkiranreddyy@gmail.com |
| S P Vimal | vimalsp | Harish Sharma | harish.sharma |
| Sunita Singhal | sunita_bansal | Nitin Rajput | nitinsinghrajputt@gmail.com |
| Vikas Singh | vikas | | |
| Virendra S Shekhawat | vsshekhawat | | |

**1. Objective:**

The primary goals of the course are to introduce:

- Basic representation of data and how to process data using the representation inside a computer.
- Techniques for specifying data, operations on data, and problem solving using a programming language.
- Systematic techniques and approaches for constructing programs.

**2. Scope:**

The course covers the following topics: Basic Model of a Computer; Problem Solving – Basic Computing Steps and Flow Charting (Assignment, Sequencing, Conditionals, Iteration). Programming Constructs – Expressions, Statements, Conditionals, Iterators/Loops, Functions/Procedures; Data Types – Primitive Types, Tuples, Choices (Unions or Enumerations), Lists/Arrays, Pointers and Dynamically Allocated Data. Input output and Files.

While the topics are taught using a specific language, the intent of the course is to teach a programming methodology, and not a programming language. There is also a laboratory component that involves development and testing of iterative and procedural programs using bounded and unbounded iterations, function composition, random access lists, sequential access lists, dynamically allocated lists, and file access.

**3. Text and Reference:**

**3 (a) Text Book:**

**TB1.** J.R. Hanly and E.B. Koffman, *Problem Solving and Program Design in C*. 5th Edition. Pearson Education 2007.

**3 (b) Reference Books:**

**RB1.** Yale Patt, Sanjay Patel. *Introduction to Computing Systems: From bits & gates to C & beyond*, Second edition, McGraw Hill.

**RB2.** Behrouz A Forouzan & Richard F Gilberg . *Computer science A structured programming approach using C*. Cengage Learning 3rd Edition

**RB3.** Brian W. Kernighan, Dennis Ritchie. *The C Programming Language*. Prentice Hall. 2nd Edition.

**4. Teaching Pedagogy**

To improve upon the teaching pedagogy, the course will be offered using *flipped classroom*. It is a form of blended learning in which students will learn new content online by watching video lectures, and what used to be homework (assigned problems) will now be done in class with instructor offering more personalized guidance and interaction with students, instead of lecturing.

The content is recorded by Instructor-Incharge/Instructor(s) and will be available on portal https://bitspilani.coursera.org/. Coursera will serve as a platform for making online content available to the students.

Other than recorded content, for each student, each (logical) week will have following interactive sessions:

1. **Two tutorial sessions (50 mins each)**: Each session will focus on problem solving based on online available recorded content. ***Students are expected to listen to recorded lectures before coming to tutorials*** and solve problems under the supervision of instructor.
2. **One lab session (2 hours)**: Here, under the supervision of lab instructors, students will solve problems on computers.
3. **One lecture (50 mins)**: The purpose is to take any unanswered-questions/doubts of the students and not to teach any new topic.

## 5. Course Plan

### 5(a): Modules

| **Module** | **Theme** | **Learning Objectives** |
|---|---|---|
| I | Basic data, data types, and data representation | To understand how to define, represent, and process basic data. |
| II | Analyzing, designing, and managing a process or program for any given problem. | To diagrammatically understand and visualize an algorithm using boxes of various kinds. This representation gives a step by step insight to a solution of a given problem. |
| III | Basic Problem Solving – Structured Programming | To understand constructs of structured programming including conditionals and iterations |
| IV | Advanced Problem Solving – Program Structuring and Structured Data | To understand how to structure complex data and how to systematically structure large programs |
| V | User Defined Data and Dynamic Data | To understand how users can define the structure and operations of new forms of data using known forms |
| VI | Advanced Topics – File I/O and Recursion | To understand recursive programming and to understand how to access files and contents of files |

**5(b) Tutorial Lecture, Recorded lecture schedule [Legends: RL- Recorded Lecture; T - Tutorial; TB - Text Book; RB - Reference Book (as mentioned in 3(b) above)]**

| Week Number | Module Number | Topic | Recorded Lecture Number | Reference | Tutorial Number |
|---|---|---|---|---|---|
| **1** | | Introduction to the subject; teaching pedagogy | Intro_lec | | Self Learning |
| | I (a) | Basics of Computing – Data and Computation. Model of a computer | RL 1.0 | **RB1:** 1.1 to 1.7 | Self Learning |
| | I (b) | Binary number system; what data to represent in binary number system? | RL 1.1 | **RB1:** 2.1, 2.2.1 | T1 |
| | I (c) | Unsigned Integers: Binary representation; word size; extension; range. | | | |
| | I (d) | Signed Integers: Binary representation in (a) Sign magnitude, (b) one's complement, and (c) two's complement. | RL 1.2 | **RB1:** 2.2.1, 2.3 | |
| | I (e) | Advantages, disadvantages, sign extension, and range of signed integers in all three respective representations. | | | |
| | I (f) | Overflow in two's complement form. | RL 1.3 | **RB1:** 2.5.3 | |
| | I (g) | Floating point number representation | RL 1.4 | **RB1:** 2.7.2 | T2 |
| | I (h) | Character, string, and other data representation. | RL 1.4 | **RB1:** 2.7.3 | |
| | I (i) | Conversions among decimal, binary, hexadecimal, and octal number systems (for signed, unsigned, and floating point numbers) | RL 1.5 | **RB1:** 2.4, 2.7.4 | |
| | I (j) | Arithmetical and logical operations on | RL 1.6 | **RB1:** 2.5, | |

| | | | | | |
|---|---|---|---|---|---|
| | | binary numbers | | 2.6 | |
| | | | | | |
| **2** | II (a) | Flow Charts: Graphical Symbols, Examples | RL 2.1 | Recorded Lecture | T1 |
| | II (b) | Flow Charts: Sequential and Conditional constructs | RL 2.2 | Recorded Lecture | |
| | II (c) | Flow Charts: Loops; Searching and Sorting | RL 2.3 | Recorded Lecture | T2 |
| | | | | | |
| **3** | III (a) | Memory and Variables – Locations, Addresses, Definitions and Declarations; Program structure | RL 3.1 | **TB1:** 2.1, 2.2, 2.3, 2.4 | T1 |
| | III (b) | Data types, Operators, and expressions | RL 3.2 | **TB1:** 2.5 | |
| | III (c) | Enumerated Data Types | RL 3.2.1 | **TB1:** 7.3 | |
| | III (d) | Expression evaluation: Operator Precedence and Associativity | RL 3.3 | **TB1:** 2.5 | T2 |
| | III (e) | Expression evaluation with different data types: type conversion (implicit and explicit) | RL 3.4 | **TB1:** 2.5 | |
| | | | | | |
| **4** | III (f) | Logical expressions and evaluation | RL 4.1 | **TB1:** 4.1, 4.2 | T1 |
| | III (g) | Sequential and Conditional execution; Control Statements (different forms of conditional statements). | RL 4.2 | **TB1:** 4.3 | |
| | III (h) | Conditional statements: different forms of nested conditional constructs. | RL 4.3 | **TB1:** 4.7 | T2 |
| | III (i) | Multi way branching: selection control | RL 4.4 | **TB1:** 4.8 | |

| | | | | | |
|---|---|---|---|---|---|
| | | mechanism using switch statement | | | |
| | | | | | |
| **5** | III (j) | Problem solving using Iterative Constructs: While statement | RL 5.1 | **TB1:** 5.1, 5.2 | T1 |
| | III (k) | Problem solving using Iterative Constructs: For statement | | **TB1:** 5.4, 5.5 | |
| | III (l) | Problem solving using Iterative Constructs: Do-while statement | RL 5.2 | **TB1:** 5.8 | |
| | III (m) | Problem solving using Nested iterative constructs | RL 5.3 | **TB1:** 5.6, 5.7 | T2 |
| | | | | | |
| **6** | IV (a) | Structure data: One dimensional random access lists: declaring, initializing, and accessing list elements | RL 6.1 | **TB1:** 8.1, 8.2, 8.3 | T1 |
| | IV (b) | Structure data: Multi-dimensional random access lists: declaring, initializing, and accessing array elements | RL 6.2 | **TB1:** 8.7 | |
| | IV (c) | Searching: Linear and Binary search | RL 6.3 | **TB1:** 8.6 | T2 |
| | IV (d) | Sorting: selection and bubble sort | RL 6.4 | **TB1:** 8.6 | |
| | | | | | |
| | | | | | |
| **7** | IV (e) | Modularity and reuse: Functions with return types/without return types, arguments/without arguments | RL7.1 | **TB1:** 3.1, 3.4, 3.5 | T1 |
| | IV (f) | Parameter passing: Pass by value, scope of data | RL 7.2 | **TB1:** 6.1, 6.2, 6.3, 6.4 | |
| | IV (g) | Transforming week #4 programs using | RL 7.3 | Recorded | |

| | | | | | |
|---|---|---|---|---|---|
| | | functions | | Lectures | |
| | IV (h) | Transforming week #5 programs using functions | RL 7.4 | Recorded Lectures | |
| | IV (i) | Transforming week #6 programs using functions | RL 7.5 | Recorded Lectures | T2 |
| | | | | | |
| 8 | IV (j) | Data type: Pointers, referencing a memory location and obtaining a value stored at that location | RL 8.1 | **TB1:** 6.1 | T1 |
| | IV (k) | Random access lists and pointers | RL 8.2 | Recorded Lectures | |
| | IV (l) | Pointer arithmetic | RL 8.3 | Recorded Lectures | T2 |
| | | | | | |
| 9 | IV (m) | Structured data: strings | RL 9.1 | **TB1:** 9.1, 9.2, 9.3, 9.4 | T1 |
| | IV (n) | Standard Library string functions: design and implementation | RL 9.2 | **TB1:** 9.5, 9.6, 9.7 | |
| | V(a) | User defined data type: structure | RL 9.3 | **TB1:** 11.1, 11.2, 11.3 | T2 |
| | | | | | |
| 10 | V (a) | Memory Layout – Implicit vs. Explicit Allocation; Static vs. Dynamic Allocation; Motivation for Dynamic Allocation | RL 10.1 | **TB1:** 14.1, 14.2 | T1 |
| | V (b) | Linked Lists: Node structure | RL 10.2 | **TB1:** 14.3 | |
| | | Linked Lists: Inserting a node | RL 10.3 | **TB1:** 14.4 | T2 |
| | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| **11** | V (c) | Linked Lists operations: Deleting a node | RL 11.1 | **TB1:** 14.4 | T1 |
| | V (d) | Linked Lists operations: Searching | RL 11.2 | **TB1:** 14.4 | T2 |
| | | | | | |
| **12** | VI (a) | Files and File I/O: External Storage, Files and File Systems; File Operations and I/O Operations; | RL 12.1 | **TB1:** 12.1, 12.2 | T1 |
| | VI (b) | Divide and Conquer – Design using Recursion ; Recursive procedures; Recursion vs. Iteration | RL 12.2 | **TB1:** 10.1, 10.2, 10.3, 10.4 | T2 |
| | | | | | |

### 6. Evaluation Scheme: [Legends: OB - Open Book, CB - Closed Book]

| S. No. | Component | CB/OB | Time | Weight | Date |
|---|---|---|---|---|---|
| 1 | Midterm test | OB | 90 mins | 22 % | As per ID |
| 2 | Online Test | OB | 120 mins | 20 % | April 20, 2014 |
| 3 | Evaluated Tutorials | CB | 15 mins each | 18 % | Every Week (mostly) |
| 4 | Labs | OB | During Lab hour | 10 % | Every Week |
| 5 | Comprehensive Exam | OB | 120 mins | 30 % | May 07, 2014 |

### 7. Make-up Policy:

- Out of **N** number of quiz components (Evaluated Tutorials) conducted in tutorials, **(N-1)** will be considered for final grading. Similarly, out of **M** number of Lab components, **(M-1)** will be considered for final grading. No Make-up will be granted for Quizzes and Labs under any condition.
- Prior Permission of the Instructor-in-Charge is required to get make-up for the mid-term test and online test. Only on producing documentary proof of possible absence, which proves that student would be physically unable to appear for the test/exam, the decision of granting the make-up will be taken.

- Prior Permission of Dean, Instruction Division is usually required to get make-up for the comprehensive exam.
- Instructor / Dean's decision in the matter of granting Make-up would be final.

**8. Consultation Hours**: See course website

**9. Notices:** All notices concerning this course will be displayed under the ANNOUNCEMENTS section on the course website (***https://bitspilani.coursera.org/introprogramming-001***). Optionally, if there is a need, email would be used on short notice – only BITS Pilani mail would be used.

**VISHAL GUPTA**
**Instructor –In- Charge**
**CS F111(Computer Programming)**