SGKGOVERNMENT DEGREE COLLEGE
Department of Computer Science, Lesson Plan

Name of the Department : Computer Science	
Name of the Lecturer : D.Siva Phanindra	
Group / Course : I B.Sc(comp)., Semester I	
Paper : Programming in c	
Name of the Topic	introduction to Algorithms and programming languages
Hours Required	16
Learning Objectives	Algorithm – Key features of Algorithms – Some more Algorithms – Flow Charts – Pseudo code – Programming Languages – Generation of Programming Languages – Structured Programming Language- Design and Implementation of Correct, Efficient and Maintainable Programs.
Previous Knowledge to be reminded	Computer basics,typing skills,Input and output devices
Topic Synopsis	**Algorithm** An algorithm is a well defined sequence of steps that provides a solution for a given problem, while a pseudocode is one of the methods that can be used to represent an algorithm. While algorithms can be written in natural language, pseudocode is written in a format that is closely related to high level programming language structures. **Pseudocode** Pseudocode is one of the methods that could be used to represent an algorithm. It is not written in a specific syntax that is used by a programming language and therefore cannot be executed in a computer. There are lots of formats used for writing pseudocodes and most of them borrow some of the structures from popular programming languages such as C, Lisp, FORTRAN, etc. Also, natural language is used when presenting details that are not important. Most of the algorithms are presented using pseudocode since they can be read and understood using programmers who are familiar with different programming languages. Some languages such as Pascal have syntax that is very similar to pseudocode making the transformation from pseudocode to the corresponding program code easier. Pseudocode allows to include control structures such as WHILE, IF-THEN-ELSE, REPEAT-UNTIL, FOR, and CASE,

	which are present in many high level languages.
Thrust Areas	Programming Design
Skills to be learnt by Students	Agorithms,pseudocode,flowcharts.
Examples / Illustrations	Agorithms,pseudocode,flowcharts for Addition of two numbers
Additional Inputs	Various generations of programming languages
Teaching Aids Used	Black board & Chalk
References Cited	programming in c by Balaguruswamy
Student Activity Planned after Teaching	Group discussion
Any other Activities	Assaignment

Signature of the lecturer

SGKGOVERNMENT DEGREE COLLEGE
Department of Computer Science, Lesson Plan

Name of the Department : Computer Science	
Name of the Lecturer : D.Siva Phanindra	
Group / Course : I B.Sc Semester I	
Paper :	
Paper : Programming in c	
Name of the Topic	**Introduction to c:** Introduction – Structure of C Program – Writing the first C Program – File used in C Program – Compiling and Executing C Programs – Using Comments – Keywords – Identifiers – Basic Data Types in C – Variables – Constants – I/O Statements in C- Operators in C- Programming Examples – Type Conversion and Type Casting
Hours Required	16
Previous Knowledge to be reminded	Algorithms and programming languages, flow charts.
Topic Synopsis	Data types in C Language Data types specify how we enter data into our programs and what type of data we enter. C language has some predefined set of data types to handle various kinds of data that we can use in our program. These datatypes have different storage capacities. C language supports 2 different type of data types: Primary data types: These are fundamental data types in C namely integer(int), floating point(float), character(char) and void. Derived data types: Derived data types are nothing but primary datatypes but a little twisted or grouped together like array, stucture, union and pointer. These are discussed in details later. Data type determines the type of data a variable will hold. If a variable x is declared as int. it means x can hold only integer values. Every variable which is used in the program must be declared as what data-type it is.
Thrust Areas	any programming languages like c++,java,python etc...
Skills to be learnt by Students	various data types usage
Examples / Illustrations	variable initialization using data types,operators, type casting.
Additional Inputs	programming examples.
Teaching Aids Used	L.C.D projector,Black board and chalk

References Cited	programming in c by balaguru swamy
Student Activity Planned after Teaching	exam conducted
Any other Activities	Assaignment

Signature of the lecturer

SGKGOVERNMENT DEGREE COLLEGE
Department of Computer Science, Lesson Plan

Name of the Department : Computer Science	
Name of the Lecturer : D.Siva Phanindra	
Group / Course : I B.Sc(comp)., Semester I	
Paper : Programming in c	
Name of the Topic	introduction to Algorithms and programming languages
Hours Required	16
Learning Objectives	**Decision Control and Looping Statements:** Introduction to Decision Control Statements – Conditional Branching Statements – Iterative Statements – Nested Loops – Break and Continue Statement – Goto Statement **Functions**: Introduction – using functions – Function declaration/ prototype – Function definition – function call – return statement – Passing parameters – Scope of variables – Storage Classes – Recursive functions – Type of recursion – Towers of Hanoi – Recursion vs Iteration
Previous Knowledge to be reminded	data types, operators, type casting.
Topic Synopsis	**In any programming language, there is a need to perform different tasks based on the condition. For example, consider an online website, when you enter wrong id or password it displays error page and when you enter correct credentials then it displays welcome page. So there must be a logic in place that checks the condition (id and password) and if the condition returns true it performs a task (displaying welcome page) else it performs a different task(displaying error page). Using decision control statements we can control the flow of program in such a way so that it executes certain statements based on the outcome of a condition (i.e. true or false). In C Programming language we have following decision control statements.** **1. if statement** **2. if-else & else-if statement** **3. switch-case statements** **A function is a group of statements that together perform a task. Every C program has at least one function, which is main(), and all the most trivial programs can define additional**

	functions. **You can divide up your code into separate functions. How you divide up your code among different functions is up to you, but logically the division is such that each function performs a specific task.** **A function declaration tells the compiler about a function's name, return type, and parameters. A function definition provides the actual body of the function.** **The C standard library provides numerous built-in functions that your program can call. For example, strcat() to concatenate two strings, memcpy() to copy one memory location to another location, and many more functions.** **A function can also be referred as a method or a sub-routine or a procedure, etc.**
Thrust Areas	Programming Design
Skills to be learnt by Students	programming skills using function calls and references
Examples / Illustrations	biggest of three numbers, swapping programs using 2 variables and without third variable.
Additional Inputs	sub programming concepts.
Teaching Aids Used	L.C.D projector,Black board & Chalk
References Cited	programming in c by Balaguruswamy
Student Activity Planned after Teaching	Seminar
Any other Activities	Assaignment

Signature of the lecturer

SGKGOVERNMENT DEGREE COLLEGE
Department of Computer Science, Lesson Plan

Name of the Department : Computer Science	
Name of the Lecturer : D.Siva Phanindra	
Group / Course : I B.Sc Semester I	
Paper : Programming in c	
Name of the Topic	**Arrays**: Introduction – Declaration of Arrays – Accessing elements of the Array – Storing Values in Array – Calculating the length of the Array – Operations on Array – one dimensional array for inter-function communication – Two dimensional Arrays –Operations on Two Dimensional Arrays - Two Dimensional Arrays for inter-function communication – Multidimensional Arrays – Sparse Matrices **Strings:** Introduction –Suppressive Input – String Taxonomy – String Operations – Miscellaneous String and Character functions
Hours Required	16
Previous Knowledge to be reminded	Algorithms and programming languages, flow charts.
Topic Synopsis	**Arrays** a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type. Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index. All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element. **Strings** are actually one-dimensional array of characters terminated by a null character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a null. The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at

	the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello." char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
Thrust Areas	any programming languages like c++,java,python etc...
Skills to be learnt by Students	data types, functions concepts should be thorough.
Examples / Illustrations	matrix addition,string manipulations program.
Additional Inputs	programming examples.
Teaching Aids Used	L.C.D projector,Black board and chalk
References Cited	programming in c by balaguru swamy
Student Activity Planned after Teaching	group discussion
Any other Activities	Assaignment

Signature of the lecturer

SGKGOVERNMENT DEGREE COLLEGE
Department of Computer Science, Lesson Plan

Name of the Department : Computer Science	
Name of the Lecturer : D.Siva Phanindra	
Group / Course : I B.Sc(comp)., Semester I	
Paper : Programming in c	
Name of the Topic	pointers, structures and unions
Hours Required	16
Learning Objectives	Pointer Expressions and Pointer Arithmetic – Null Pointers – Generic Pointers - Passing Arguments to Functions using Pointer – Pointer and Arrays – Passing Array to Function – Difference between Array Name and Pointer – Pointers and Strings – Array of pointers – Pointer and 2D Arrays – Pointer and 3D Arrays – Function Pointers – Array 0f Function Pointer – Pointers to Pointers – Memory Allocation in C Programs – Memory Usage – Dynamic Memory Allocation – Drawbacks of Pointers **Structure, Union, and Enumerated Data Types:** Introduction – Nested Structures – Arrays of Structures – Structures and Functions – Self referential Structures – Union – Arrays of Unions Variables – Unions inside Structures – Enumerated Data Types
Previous Knowledge to be reminded	Memory allocation.
Topic Synopsis	**Pointers are powerful features of C and (C++) programming that differentiates it from other popular programming languages like Java and Python.** **Pointers are used in C program to access the memory and manipulate the address.** **Address in C** **Before you get into the concept of pointers, let's first get familiar with address in C.** **If you have a variable var in your program, &var will give you its address in the memory, where & is commonly called the reference operator.** **You must have seen this notation while using scanf() function. It was used in the function to store the user inputted value in the address of var.** **scanf("%d", &var);** **Structures**

	Structures in C are used to encapsulate, or group together different data into one object. You can define a Structure as shown below: **struct object {** **char id[20];** **int xpos;** **int ypos;** **};** **Structures can group data of different types as you can see in the example of a game object for a video game. The variables you declare inside the structure are called data members.** **Unions** **Unions and Structures are identical in all ways, except for one very important aspect. Only one element in the union may have a value set at any given time. Everything we have shown you for structures will work for unions, except for setting more than one of its members at a time. Unions are mainly used to conserve memory. While each member within a structure is assigned its own unique storage area, the members that compose a union share the common storage area within the memory. Unions are useful for application involving multiple members where values are not assigned to all the members at any one time.** **Let us modify our structure object from above so that it has a union for indicating dead or alive in it:** **struct object {** **char id[20];** **struct coordinates loc;** **union deadoralive {** **int alive;** **int dead;** **}** **};**
Thrust Areas	Programming Design
Skills to be learnt by Students	Programming skills
Examples / Illustrations	enumerated datatypes, null pointer.
Additional Inputs	pointers- Arrays,strings,functions
Teaching Aids Used	L.C.D Projector,Black board & Chalk
References Cited	programming in c by Balaguruswamy

Student Activity Planned after Teaching	Group discussion
Any other Activities	Assaignment

Signature of the lecturer

SGKGOVERNMENT DEGREE COLLEGE
Department of Computer Science, Lesson Plan

Name of the Department : Computer Science	
Name of the Lecturer : D.Siva Phanindra	
Group / Course : I B.Sc Semester I	
Paper : Programming in c	
Name of the Topic	Introduction to Files – Using Files in C – Reading Data from Files – Writing Data from Files – Detecting the End-of-file – Error Handling during File Operations – Accepting Command Line Arguments – Functions for Selecting a Record Randomly - Remove() – Renaming a File – Creating a Temporary File
Hours Required	16
Previous Knowledge to be reminded	records,fields.
Topic Synopsis	C Programming Files I/O There are a large number of functions to handle file I/O (Input Output) in C. In this tutorial, you will learn to handle standard I/O in C using fprintf(), fscanf(), fread(), fwrite(), fseek. File Operations In C, you can perform four major operations on the file, either text or binary: Creating a new file Opening an existing file Closing a file Reading from and writing information to a file Working with files When working with files, you need to declare a pointer of type file. This declaration is needed for communication between the file and program. FILE *fptr; Opening a file - for creation and edit Opening a file is performed using the library function in the "stdio.h" header file: fopen(). The syntax for opening a file in standard I/O is: ptr = fopen("fileopen","mode") For Example: fopen("E:\\cprogram\\newprogram.txt","w"); Closing a File The file (both text and binary) should be closed after

	reading/writing. Closing a file is performed using library function fclose(). fclose(fptr); //fptr is the file pointer associated with file to be closed.
Thrust Areas	any programming languages like c++,java,python etc...
Skills to be learnt by Students	file operations
Examples / Illustrations	read,write,open,close
Additional Inputs	programming examples.
Teaching Aids Used	L.C.D projector,Black board and chalk
References Cited	programming in c by balaguru swamy
Student Activity Planned after Teaching	practiced programs
Any other Activities	Assaignment

Signature of the lecturer