|  |  |
| --- | --- |
| **SESSION** | **February/MARCH 2024** |
| **PROGRAM** | **BACHELOR of COMPUTER APPLICATIONS (BCA)** |
| **SEMESTER** | **I** |
| **course CODE & NAME** | **DCA1102; Programming in C** |
| **CREDITS** | **4** |

**Set-Ist**

**1. What is the basic structure of C program? Explain printf () and scanf () functions with an example.**

**Ans:**A basic structure in C program can be viewed as a group of building blocks called functions. A function is a subroutine that may include one or more statements designed to perform a specific task.

To write a C program we first create functions and then put them together. A C program may contain one or more sections shown in Fig.

The documentation section consists of a set of comment (remarks) lines giving the name of

Its Half solved only

Buy Complete assignment from us

**Price – 190/ assignment**

**MUJ Manipal University Complete SolvedAssignments session FEB 2024**

buy cheap assignment help online from us easily

we are here to help you with the best and cheap help

**Contact No – 8791514139 (WhatsApp)**

**OR**

**Mail us- bestassignment247@gmail.com**

**Our website -** [**www.assignmentsupport.in**](http://www.assignmentsupport.in)

**2. What is function? Explain different classification of user defined functions based on parameter passing and return type with examples.**

**Ans:**

A function is a self-contained program segment that carries out some specific, well-defined task. Every C program contains one or more functions. One of these functions must be called main. Program execution will always begin by carrying out the instructions in main. Additional functions will be subordinate to main, and perhaps to one another.

**There can be 4 different types of user-defined**[**functions**](https://www.studytonight.com/c/user-defined-functions-in-c.php)**, they are:**

1. Function with no arguments and no return value

**3. Explain the different kinds of loops available in C with examples.**

**Ans:**

**In C programming,** loops are used to repeatedly execute a block of code as long as a specified condition is true. There are three primary types of loops in C: for, while, and do-while. Each type of loop serves a specific purpose and is used in different scenarios.

**1. For Loop**

The for loop is typically used when the number of iterations is known beforehand. It consists

**Set-Iind**

**4. What is the purpose of storage class in C? Explain various types of storage class in C.**

**Ans:**The storage class in the C language for determining the visibility, lifetime, initial value, and memory location of any given variable. The storage classes define the visibility (scope) and the lifetime of any function/ variable within a C program. These classes precede the type that they are going to modify.

**There are four types of storage class in C:-**

**Automatic or local variables**

**5. Describe the process of dynamic memory allocation in C and elaborate on the different dynamic memory allocation functions in the C programming language.**

**Ans:**The process of allocating memory at run time is known as Dynamic Memory Allocation. Although C does not inherently have this facility, there are four library routines known as “Memory Management Functions” that can be used for allocating and freeing memory during program execution. They are listed in Table 12.1. These functions help us build complex

**6. a. What is the difference between structure and union in C?**

**Ans:Here’s a detailed comparison between structures and unions:**

**Structure (struct)**

**Definition:** A structure is a user-defined data type that allows the combination of data items of different kinds.

**Memory Allocation:** Each member of a structure has its own memory location. The total memory allocated to a structure is the sum of the memory required for each member.

**Accessing Members:** All members can be accessed individually and simultaneously. Each

**b. Discuss the concept of macro substitution in C preprocessor directives. Provide examples of macro definitions and demonstrate how they are expanded during compilation.**

**Ans:**

**Macro Definition and Substitution**

A preprocessor line of the form #define name text defines a macro with the given name, having as its value the given replacement text. After that (for the rest of the current source file), wherever the preprocessor sees that name, it will replace it with the replacement text. The name follows the same rules as ordinary identifiers (it can contain only letters, digits, and underscores, and may not begin with a digit). Since macros behave quite differently from