

Computer Science		PAPER: II
Time: 2.10 Hours	(SUBJECTIVE TYPE)	Marks: 60

SECTION-I

(MS-ACCESS)

2. Write short answers to any SIX (6) questions: (12)

(i) Define database.

Ans Database is a collection of logically related data sets or files. Normally, this collection possess file or datasets of different nature used for specific purposes. Different type of information is stored and later retrieved in different ways to meet different processing requirements.

(ii) Write two examples of entity.

Ans Two examples of entity are:

1. TEACHER
2. STUDENT

(iii) What is database wizard?

Ans Wizard makes very easy to create a database. You can choose from several examples of databases in the Database Wizard for such storage uses as contact information, inventory control, a ledger, and so on. You can create and then modify these databases to meet your own needs.

(iv) How referential integrity can be achieved?

Ans Referential integrity requires that a foreign key must have a matching primary key or it must be null. This constraint is specified between two tables; it maintains the correspondence between rows in these tables. It means the reference from a row in one table to another table must be valid.

(v) Define filter.

Ans A filter is used to extract the records that match a set of criteria. Filters are used with opened tables. Filter by selection and filter by form are its common types.

(vi) Differentiate between relationship and join.

Ans When you create a data source, it has two layers. The top-level layer is the logical layer of the data source. You combine data between tables in the logical layer using relationships. On the other hand, you combine data between tables at the physical layer using joints.

(vii) What is the purpose of radio button?

Ans Radio button is also known as option button. It is used to display Yes / No, True / False or On / Off option to the user. It is used when user is required to select one option from more than one option. The user can select an option by click on a radio button. De-selection is done by clicking again on it.

(viii) List two database objects used to create report.

Ans Two database objects used to create report are:

1. Tables
2. Queries

(ix) What is the use of index in database management system?

Ans Indexing is a data structure technique to efficiently retrieve records from the database files based on some attributes on which the indexing has been done.

C-LANGUAGE

3. Write short answers to any SIX (6) questions: (12)

(i) Define syntax error. Give an example.

Ans Syntax Error:

A set of rules for writing programs in a programming language is known as syntax of that language. Syntax error occurs when an invalid statement is written in a program and rules of language are not followed. The compiler can detect syntax errors. A program with this type of errors cannot be compiled successfully

Example: Typing 'pintf' instead of 'printf'.

(ii) Differentiate between unary and binary operators.

Ans Unary operator has just one operand e.g., ++, --. On the other hand, binary operator has two operands e.g., +, -, *, /.

(iii) How implicit type casting performed in C?

Ans Implicit type casting is the process of converting the data type of a value during execution automatically by the C compiler. The operands in arithmetic operation must be of similar types otherwise the value with lower data type is converted into higher data type.

```
int a;  
long b;  
c=a*b;
```

In the above expression, data type of 'a' is lower than that of 'b'. So the value of 'a' will be converted into long during the evaluation of expression. It is important that the data type of 'a' is not changed but only the data type of the value of 'a' is changed during the evaluation.

(iv) Write any two rules for declaring variables in C.

Ans Two rules for declaring variables in C language are:

1. C is a case sensitive language. Thus, the names count and COUNT refers to two different variables.
2. C keyword can't be used as variable names, e.g., we cannot use int, void, signed, or while as variable names.

(v) List any four functions for input.

Ans C language provides many functions to take input from the user. Some important functions for input are as follows:

- scanf()
- gets()
- getch()
- getche()

(vi) Write down any four escape character provided by C.

Ans Four escape characters provided by C are:

Characters	Purpose
\n	New Line
\t	Tab
\b	Backspace
\f	Form feed

(vii) Trace out the errors in the following code segment:

```
#include < stdio . h >
#include < conio . h >
Void main ( ) ;
{
printf ("Hello")
getch ( ) ;
}
```

Ans Error:
Semicolon (;) after void main (.)

(viii) Predict the output of the following code segment:

```
int x = 15;
int y = 5;
printf ("% d\t % d", x % y, x / y );
```

Ans 0 3

(ix) Predict the output of the following code segment:

```
void main ( )
{
printf ("\n \t I LOVE");
printf ("\t MY");
printf ("\t Country");
}
```

Ans I LOVE MY Country.

(C-LANGUAGE)

4. Write short answers to any SIX (6) questions: (12)

(i) Find the errors in the following code:

```
float y = 3.14
if (y == 4)
print ("Error");
Else
printf ("No Error");
```

Ans The "e" of "Else" is capital and this is the error.

(ii) Determine the output of following code:

```
char ch = 'c';  
switch (ch)  
{  
case 'c':  
printf("A");  
case 'd':  
printf ("B");  
}
```

Ans The output will be A.

(iii) Write the output of following code:

```
int x = 25;  
(x % 2 == 0? printf ("Even"): printf ("Odd");
```

Ans Odd.

(iv) Differentiate between while-loop and do-while loops.

Ans While loop is the condition which comes before body of loop. Body of loop executed if condition is true. On the other hand, do-while loop is the condition after body of loop. Do-while loop body executed at least once if condition is false.

(v) Define infinite loop.

Ans An infinite loop in C-language is a looping construct that does not terminate the loop and executes the loop forever. It is also called an endless loop.

(vi) What is function declaration?

Ans A function declaration tells the compiler about a function's name, return type and parameters.

Syntax

return type function name (parameter list)

```
{  
Body of the function  
}
```

(vii) Write benefits of using functions.

Ans Benefits of using functions are:

1. With functions, the program becomes simple as it is divided into smaller and independent units or modules.
2. Writing and testing of program becomes easier.
3. Several programmers can work on one program simultaneously and develop a program quickly.
4. Re-usability of function is important benefit.
5. Use of functions also reduces overall length of the program.

(viii) Write about binary stream.

Ans Binary Stream:

A binary stream is a sequence of bytes. The translation is not needed in binary stream. It exists with one-to-one correspondence.

(ix) What is the use of formal parameters?

Ans The parameters specified in the function header are called formal arguments or formal parameters of the function.

The function uses its formal parameters for processing data passed to it. Any change made to the value of formal parameters does not affect the value of actual parameters.

SECTION-II

(MS ACCESS)

Note: Attempt any ONE (1) question.

-
5. What is Database Management System? Discuss its objectives. Also write down disadvantages of Database Management System. (8)
-

Ans Database Management System:

The data management system (a collection of programs) which is used for storing and manipulating databases is called database management system (DBMS). It is an improvement over the traditional file management systems. It uses DBMS software (database manager) which controls the overall structure of a database and access to the data itself.

Normally, the DBMS is used for large- or medium-sized organizations, having heterogeneous types of files, used for different purposes. In this mechanism, the data elements are so integrated, cross referenced and shared amongst them that the duplication of data is almost impossible.

Objectives of Database Management System:

- **Shareability:**

Different people and different processes must be able to use the same actual data virtually the same time.

- **Availability:**

Both the data and the DBMS which delivers the data must be easily accessible to the users.

- **Evolveability:**

The ability of the DBMS to change in response to growing user needs and advancing technology.

- **Database Integrity:**

Since data is shared among multiple users, adequate integrity control measures must be maintained.

Disadvantages of Database Systems

1. Require Additional System Overhead:

Additional overhead is required to access data, in case of doing some simple jobs; like reading and processing a tape file, which might take a little time and resources to do the job. If we have to do it on DBMS, it is like "requiring too much to do too little".

2. Additional Training Required for Training of Staff:

Application programmers required a sort of precise training to code efficient programs that will run under a DBMS. There is a possibility that in adequate training or experience of application development staff will lead the creation of grossly inefficient database calls. Quite often, the problem might not be found until the program reaches production status. The typical example is that of using proper and improper indexes for accessing the database.

3. Problems can Multiply in Selecting a Wrong Type of Dbase Environment:

A later change in structure, forced by changing requirements, can be costly in terms of conversion and testing of existing programs. Hierarchical data base systems are, in particular, more sensitive than network or relational systems towards this kind of problem, and implementing changes costs a great deal. On the other hand, doing these changes on relational data bases are fairly easy and less costly.

4. Data Must be Considered a Corporate Resource:

The data in a company's data base no longer belong to one organization alone. True, one organization normally has the primary responsibility for creating a data base. However, as data base systems mature, more companies or organizations can share the same data across applications.

5. A Need of a Dictionary:

In order to share data across application systems, or to simply given end users, the ability to identify the location of information they need in order to do their jobs, the internal data contents of a company's data bases need to be documented in a consistent manner. For this purpose, they have to install a data dictionary system, which is another overhead on the DBMS.

6. What is normalization? Explain third normal form with an example. (8)

Ans Normalization:

Normalization is the process of converting complex data structures into simple and stable data structures. It is based on the analysis of functional dependence.

In other words, Normalization is a technique for reviewing the entity / attribute lists to ensure that attributes are stored "where they belong". It is the basis for a relational data base system. In practice, it is simply an applied common sense. More formally stated, it is the process of analyzing the dependencies

of attributes within entities. Attributes for each entity are checked consecutively against three sets of rules, making adjustments when necessary to put the entity in First, Second and Third normal form.

Third Normal Form (3 NF):

A relation is in third normal form (3 NF) if it is in 2 NF and no transitive dependencies exist.

A more precise definition for 3 NF is: "A non-key attribute must not depend on any other non-key attribute" or if a non-key attribute's value can be obtained simply by knowing the value of another non-key attribute, the relation is not in 3 NF.

Example:

Consider a relation as follows:

SALES(CUSTNO,NAME,SALESMAN,REGION)

Where CUSTNO is the primary key.

The following functional dependencies exist in the relation.

(a) CUSTNO -----> NAME,SALESMAN

(b) SALESMAN -----> Region (since each salesman is assigned a unique region)

Notice that SALES is in 2 NF, because the primary key consists of a single attribute (CUSTNO). However, there is a transitive dependency, because REGION is functionally dependent on SALESMAN which in turn is functionally dependent on CUSTNO. As a result, there are update anomalies in relation SALES.

CUSTNO	NAME	SALESMAN	REGION
8023	AAAA	Ahmad	South
9167	BBBB	Bashir	West
7924	CCCC	Ahmad	South
6837	DDDD	Khalid	East
9596	EEEE	Bashir	West
7018	FFFF	Munir	North

A relation with Transitive dependency

SECTION-III

Note: Attempt any TWO (2) descriptive answers of the following questions.

(C-LANGUAGE)

7. Briefly describe the basic structure of C Program. (8)

Ans Basic Structure of a C Program:

The structure of a C program is very flexible which increases the power of the language. C is a structured programming language; therefore, it provides a well-defined way of writing programs. To understand the basic structure of the C program, we proceed with the following example:

- (i) In unstructured programming languages, the entire logic of the program is implemented in a single module (function), which causes the program error prone, difficult to understand, modify and debug.
- (ii) In structured programming languages, the entire logic of the program is divided into number of smaller modules, where each module (piece of code) implements a different functionality.

Hello World – A simple C program:

Let us consider a simple C program that displays the phrase Hello World! on the screen.

```
#include <stdio.h>
void main (void)
{
    printf("Hello World!");
}
```

The above Hello World program has two parts:

- The processor directive (#include <stdio.h>)
- The main function

Preprocessor Directives:

Preprocessor directives are commands that give instructions to the C preprocessor. The preprocessor is a program that modifies the C program (source program) prior to its compilation. A preprocessor directive always begins with the symbol (#). In the above program, include is a preprocessor directive.

The **include** directive gives a program access to a library. This directive causes the preprocessor to insert definitions from a standard header file into a program before compilation. Hence, the statement `#include<stdio.h>` gives the program access to standard input and output functions.

#include Directive for Defining Identifiers from Standard Libraries

SYNTAX: `#include<standard header file>`

EXAMPLE: `#include <stdio.h>`
`#include <math.h>`

Another important preprocessor directive is `#define` directive. It is used to define a constant macro. Examples of this macro will be discussed in subsequent chapters.

define Directive for Defining Constant Macros

SYNTAX: `#define Macro_Name expression`

EXAMPLE: `#define PI 3.142857`
`#define SEC_PER_HR 3600`

Function main:

As shown in the above Hello World program, the definition of the main function comes next to the specification of the `#include` preprocessor directive. In fact, `main` is the function where the execution of the C program begins. Every C program has a main function. The rest of the lines of program forms the body of the main function, the body is enclosed in braces `{` and `}`.

C programs are divided into units called functions. This division is usually done on the basis of functionality, where every function carries out a single task. However, it is not necessary to divide every program into functions. The same functionality may be achieved through a single function. But, every C program must have the function `main` as the execution of the program starts from there. In this way, we can say that the main function is actually the entry point of the C programs.

main Function Definition:

SYNTAX: `void main (void)`
`{`
`body of main function`
`}`

8. Define "if-else-if" statement. Explain its working with syntax, flowchart and example. (8)

Ans if-else if Statement:

Nested *if* statements can become quite complex, if there are more than three alternatives and indentation is not consistent, it may be difficult to determine the logical structure of the *if* statement. In such situations, *if* statement with multiple alternatives (*if-else if*) can be a good option.

Syntax:

The general form of *if-else if* statement is as follows:

```
if (condition1)  
    statement1;  
else if (condition2)  
    statement2;
```

```
    .  
    .  
else if (conditionn)  
    statementn;  
else  
    statementk;
```

The test conditions in *if* statement with multiple alternatives are executed in sequence until a *true* condition is reached. If a condition is true, the statement(s) following it is executed, and the rest of the alternatives are skipped. If a condition is false, the statement following it is skipped and the next condition is tested. If all conditions are false, then statement_k following the last *else* is executed.

Flowchart:

The following flowchart shows the execution flow of the program through *if-else if* statement.

Example:

The program in example can be re-written using *if-else if* structure as follows:

```
#include <stdio.h>
void main (void)
{
    Int num;
    printf ("Enter a number >");
    scanf ("%d", &num);
    if (num > 0)
        printf ("The number is positive");
    else if (num < 0)
        printf ("The number is negative");
    else
        printf ("The number is zero");
}
```

It can be seen that the *else-if* construct has greatly simplified the program, while preserving the efficiency as well. If a positive number is entered, we will reach the answer in the first comparison and rest of the conditions will be skipped.

9. Write a program in C-language to print digits from 1 to 100. (8)

Ans ▶ # include <stdio.h>
void main (void)
{
 int count;
 count = 1;
 while (count <= 10)
 {
 printf ("%d\n", count);
 count = count + 1;
 }
}