THE QUAIDE MILLETH COLLEGE FOR MEN
(Reaccredited by NAAC)
INTERNAL QUALITY ASSURANCE CELL

THE QUAIDE MILLETH COLLEGE FOR MEN

Medavakkam, Chennai-600100.

B.Com(ISM)

PROGRAMMING IN C

MAT3B

Question Bank and Answer Key

UNIT – I

Part - A

1. **Define identifiers.**

   An identifier is a name given to the program elements such as variables, functions, procedures, arrays etc. It is a symbolic name given to the memory.

2. **Write any two rules for identifiers.**

   - The identifier should contain alphabets.
   - White spaces are not allowed.

3. **Define reserved word.**

   Reserved word is having a pre-defined meaning. Like,

   auto, break, do, goto, while etc.

4. **Define variable.**

   A variable is an identifier, which changes its value during the execution of the program. It is a symbolic name given to the memory location.

   datatype var1,var2,…….;

5. **How do you declare the variables in C?**

Syntax:

datatype var1,var2,…….;

Example:

int i,n,sum=0;

6. **Define constant.**

Constant does not change its value during execution. The types of constants are Integer, Floating-point, Character and String constants.

7. **Write the use of '\n'.**

This is one of the escape sequence, used to set new line in the output. It represents non-printing characters.

8. **Write the syntax and purpose of ternary operator.**

Ternary operator (?:) also known as conditional operator. Operates on three operands. General form is

test-expression ? true-expression : false-expression;

9. **Define expression.**

Expression is defined as the combination of constant, variable, operator etc. There are Arithmetic, Relational expressions etc.

10. **Define statement.**

A statement is a command given to the Computer that instructs the Computer to take a specific action, such as display to the screen, or collect input. A Computer Program is made up of a series of statements.

Part - B

1. Explain the structure of C program.

```
include files
directives declaration
global declaration
main()
{
    variable declaration
    …
    …
    program statements
    …
    …
}
sub-function statements
```

Following is the example program that illustrates how C program can be written.

Example program

2. Discuss about different types of constant.

Constant does not change its value during execution. The types of constants are

i) Integer constant: An Integer constant can hold only integer values.

ii) Floating point constant: A Floating-point number can contain a decimal number or exponent.

iii) Character constant: A character constant consists of only one character enclosed within single quotes.

iv) String constant: A string constant consists of a sequence of characters.

3. Define Expression. Write about the different types of Expression.

Expression is defined as the combination of constant, variable, operator etc. There are Arithmetic, Relational expressions etc.

i) Arithmetic expression: Expression which uses arithmetic operator.

(A+B)*C

ii) Relational expression: Relational operator used as conditional expression.

a>b

iii) Logical expression: Logical operator used in the conditional expression.

(a>b) && (a>c)

4. Write in detail about data types in 'C'.

Datatypes in C classified into: 1. Fundamental and 2. Derived datatypes.

Fundamental datatypes used to represent actual data in the memory. It is the basic datatype.

i) int datatype: Integer numbers represented using int.

ii) float datatype: Floating-point numbers represented using float datatype.

iii) double datatype: Double-precision floating–point numbers

iv) char datatype: to represent a single character or a group of characters depending upon the size that is declared.

Part - C

1. Describe in detail, about the different types of operators.

C includes a set of operators that act upon the given operands to produce a value.

Unary operator:

i) Unary Minus:

A minus sign precedes a numeric constant.

ii) Increment operator:

This operator is used to increase the value of the operand by one.

++i (pre-increment operator)

i++ (post-increment operator)

iii) Decrement operator:

This operator is used to decrease the value of the operand by one.

--i (pre-decrement operator)

i-- (post-decrement operator)

iv) Sizeof:

sizeof operator is used to return the size of operand, in bytes.

Binary operator:

i) Arithmetic operator:

Used to calculate the values.

Operators: + - * / %

ii) Relational operator:

Used to check given two operands.

Operators: < <= > >= == !=

iii) Logical operator:

Used to make comparison between two or more conditions.

Operators: && || !

iv) Assignment operator (=):

Used to assign a value to the variable. Value of right-hand side is stored to the left-hand side variable.

- Simple Assignment operator
- Compound Assignment operator

Ternary operator:

This operator is known as Conditional operator (?:). This operator simplifies the if…else statement. Operates on three operands. General form is

test-expression ? true-expression : false-expression;

UNIT – II

Part - A

1. **Mention any two formatted functions in C.**
2. **Write the syntax of 'scanf' and 'printf' statements.**

scanf:

Syntax: scanf(control string, argument list);

printf:

Syntax: printf(control string, argument list);

3. **Write the purpose of goto statement.**

goto statement is the unconditional control statement. It transfers the control from one place to another by specifying the label.

Syntax: goto label;

4. **What is the purpose of a break statement?**

Break statement is used to terminate the loop. It can be used in the switch statement.

Syntax: break;

5. **Give the syntax of if…else statement.**

```
if(conditional expression)
  statement-1;
else
  statement-2;
```

6. **Write the syntax of nested if…else statement.**

```
if(conditional expression-1)
    if(conditional expression-2)
          statement-1;
    else
          statement-2;
```

```
else
   if(conditional expression-3)
        statement-3;
   else
        statement-4;
next statement
```

Part - B

1. Explain the syntax and purpose of ‘scanf’ and ‘printf’ statements in C.

scanf:

Input data can be entered into the Computer from a standard input device by means of the C library function scanf(). This function can be used to enter any combination of numerical values, single character values and strings.

Syntax:

```
scanf(control string, argument list);
```

The control string containing certain required formatting information. In argument list, variable name is given to store the value.

printf:

This function is used to display the data on the standard output device i.e. Monitor.

Syntax:

```
printf(control string, argument list);
```

The control string gives the format of data to be displayed on the Monitor. The argument list contains array name, variables, constants to be printed on the output device.

2. Explain about if-else statement with example.

if-else statement is used when the statement contains one condition and two statements.

Syntax:

```
if(conditional expression)
        statement-1;
else
        statement-2;
```

If the given condition becomes true then the statement-1 will be executed, then the control jumps to the next statement. If the condition becomes false, statement-2 will be processed.

Example

Flowchart

Following example program shows the concept of if-else statement.

Example program

3. Describe about nested if-else statement.

If an if-else statement uses another if-else statement then it is called nested if-else statement.

Syntax:

```
if(conditional expression-1)
   if(conditional expression-2)
        statement-1;
   else
        statement-2;
else
   if(conditional expression-3)
        statement-3;
   else
        statement-4;
next statement
```

If the given condition-1 becomes true then the condition-2 will be tested and if the condition-1 becomes false, condition-3 will be tested.

Example

Flow chart

Following program illustrates the use of nested if-else statement.

Example program

4. Explain the syntax of ‘switch…case’ statement with an example.

The switch-case statement is as like as the if-else statement. With the help of switch statement, the branching can be extended to any number.

The general form is

```
switch(expression)
{
    case label-l:
            statement-1;
            break;
    case label-2:
            statement-2;
            break;
    ……………
    ……………
    case label-n:
            statement-n;
            break;
    default:
            default statement;
}
next statement;
```

When this statement is executed, the Computer first evaluates the value of the expression in the keyword switch. The value is successfully compared with the cases such as case label-1, label-2, ….. label-n. If a case label matches with the value, the statement block associated with the case label is executed. Then the control is transferred to the next statement. If none of the cases matches with the value, the default statement block is executed.

Example

Flow chart

The use of switch-case statement is shown in the following program.

Example program

5. Summarize about while loop with example program.

While loop is the entry-controlled loop. If the given condition is true, the given statements will be executed. Again and again the condition is tested, the given statements will be executed repeatedly. The control will comes out of the loop only when the condition becomes false.

Syntax:

```
while(conditional expression)
{
    statement;
}
```

Example

Flow chart

Following is the example program to illustrate the while loop.

Example program

6. Explain in detail about do-while loop.

do-while loop is the exit-controlled loop. The control enters directly into the loop. If the given condition is true, the given statements will be executed. Again and again the condition is tested, the given statements will be executed repeatedly. The control will comes out of the loop only when the condition becomes false.

Syntax:

```
do
{
    statement;
} while(conditional expression);
```

Example

Flow chart

Following program illustrates the do-while loop concept.

Example program

Part - C

1. **Discuss about various Input and Output functions in C with suitable examples.**

Input function:

getchar:

To get a single character. The argument for this function is single character.

getc:

Similar to getchar() function. This function is used for files instead of transferring the input data to the monitor.

gets:

gets() function is used to accept a string.

scanf:

Input data can be entered into the Computer from a standard input device by means of the C library function scanf(). This function can be used to enter any combination of numerical values, single character values and strings.

Syntax:

scanf(control string, argument list);

The control string containing certain required formatting information. In argument list, variable name is given to store the value.

Output function:

putchar:

To get a single character on the standard output device.

putc:

Similar to putchar() function. The data from the file is displayed.

puts:

puts() function is used to display the string. Here, the argument is the string.

printf:

This function is used to display the data on the standard output device i.e. Monitor.

Syntax:

```
printf(control string, argument list);
```

The control string gives the format of data to be displayed on the Monitor. The argument list contains array name, variables, constants to be printed on the output device.

2. Describe about for loop in detail.

for loop is the entry-controlled loop. In for loop, there are four parts. for loop processes all the four parts.

i) Initialization

ii) Conditional expression

iii) Increment or decrement part

iv) Statement

Syntax:

```
for(initialization;conditional expression;increment)
{
    statement;
}
```

i) When enters the loop, first initializes.

ii) The condition will be tested.

iii) Statement will be executed, if the condition becomes true.

Example

Flow chart

Following is the example program to show the use of for loop.

Example program

3. Discuss various conditional looping constructs available in C.

while loop - definition, syntax, example.

do-while loop - definition, syntax, example.

for loop - definition, syntax, example.

UNIT – III

Part - A

1. **Define function.**

   Functions are the building blocks of C program. Functions are one of the part of the program. A function can be invoked from different parts of the main() program, by calling it. This function is called user-defined function.

2. **How to access function in C?**

   Below given is syntax of using the function defition.

   ```
   return-type fun-name(parameter- list)
   {
      statement
   }
   ```

3. **What do you mean by function prototype?**

   A function prototype that is the function declaration, declares a function before it is used and prior to its function definition. Function prototype consists return type, function’s name, parameter list and semi-colon.

4. **What is the use of return statement?**

   A return statement is used to return a value to the calling program.

   Syntax:

   ```
   return [value];
   ```

5. **What is meant by recursion?**

   Recursion is the type of function which calls by itself. When there is the logical situation appears, the function returns to main function.

6. **What is Local variable?**

   The variable declared within a function is called a variable i.e. the Local variable. The value of the local variable is valid only within the given function.

7. **What do you mean by Global variable?**

   The variables which are declared outside the functions are called Global variable. These variables can be used by all the functions of the program.

8. **What is the use of auto variable?**

   Without using the auto keyword, the variable declaration is the auto storage class by default, the local variable is the auto variable.

9. **Define static variable.**

   The value of static variable can be retained upto the end of the program. Both global and local variables can be declared as static.

10. **What do you mean by register variable?**

    Registers are special storage areas placed in the CPU. A local variable or formal parameter can be declared to be register.

11. **Define extern variable.**

    The extern storage class is created only when the variable is declared as global. The value can be retained throughout the execution of the program.

12. **How typedef can be used?**

    This storage class is used to create new datatype that are equivalent to the existing datatype.

    ```
    typedef datatype new-type;
    new-type var1,var2,….;
    ```

Part - B

1. **Define recursion. Explain in detail.**

   Recursion is the type of function which calls by itself. When there is the logical situation appears, the function returns to main function. A condition should be provided within the recursive function.

A recursive program can be illustrated diagrammatically explained as follows.

Diagram

Following example program illustrates the concept of recursion.

Example program

2. **How will you pass an argument to a function? Explain.**

To pass an argument to a function, there are two types of arguments.

1. Formal argument
2. Actual argument

Formal argument:

A name in a function or subroutine definition that is replaced by, or bound to, the corresponding actual argument when the function or subroutine is called.

Actual argument:

Actual arguments and formal arguments, they are called "arguments" because they define information that is passed to a function. Actual arguments are arguments as they appear in function calls. Formal arguments are arguments as they appear in function declarations.

```
return-type fun-name(parameter- list)
{
   statement
}
```

Following program illustrates the concept of passing arrays to a function.

Example program

Part - C

1. **Give a brief note on function.**

Functions are the building blocks of C program. Functions are one of the part of the program. A function can be invoked from different parts of the main() program, by calling it. This function is called user-defined function.

Syntax:

```
return-type function-name(argument-list)
{
    body of the function
}
```

Example

There are 3 parts in the user-defined function.

1. Function header
2. Body of the function
3. Function prototype

A function prototype that is the function declaration, declares a function before it is used and prior to its function definition. Function prototype consists return type, function's name, parameter list and semi-colon.

Diagram

Following program shows the concept of function.

Example program

2. **Discuss about various storage classes in C.**

Storage class refer to its scope within the program that the limitation of the variable.

The syntax is

storage-class datatype var1,var2,var3,…..;

Example:

static int a,b,c;

There are five storage classes available in C language. They are

a. auto variable

Without using the auto keyword, the variable declaration is the auto storage class by default, the local variable is the auto variable.

auto i,n,sum=0;

b. static variable

The value of static variable can be retained upto the end of the program. Both global and local variables can be declared as static. Suppose, a variable is declared in a function, value of that variable will not be closed till the end of the program. Formal parameters cannot be declared as static.

```
static int c=0,z;
```

c. register variable

Registers are special storage areas placed in the CPU. A local variable or formal parameter can be declared to be register. Declaring variables as register increases the execution speed. However, only integer variables can be declared as register.

d. extern variable

The extern storage class is created only when the variable is declared as global. The value can be retained throughout the execution of the program. This storage class does not caused memory to be reserved for the variable.

e. typedef

This storage class is used to create new datatype that are equivalent to the existing datatype.

```
typedef datatype new-type;
new-type var1,var2,….;
```

where datatype refers to the existing basic datatype and the new-type refers to the new datatype defined by the user.

UNIT – IV

Part - A

1. **Define array.**

   An array is a group of same type that share a common name. The array number are enclosed within square brackets.

   ```
   storage-class datatype array-name[size];
   ```

2. **What is an unsized array?**

   The array need not to be specified, if the array size is not given, the compiler count the number of values as size.

3. **What is two-dimensional array?**

   A two-dimensional array contains two subscripts. A two-dimensional array can be considered as a table made up of rows and columns and can be declared as x[i][j].

4. **Define string.**

   Characters can be stored in arrays. String is stored as a sequence of characters given with double quotes.

5. **Define structure.**

   A structure is a type of data structure whose individual element can differ in type. The elements of the structure are members.

6. **Write the syntax of structure.**

   ```
   struct tag
   {
       member 1;
       member 2;
       …
       …
       member n;
   ```

```
}
struct tag var1,var2,…;
```

7. **Define pointer.**

A pointer is a variable that holds the address of another object.

```
int *p;
```

8. **What are the operators that supports pointer?**

Two special operators to refer the memory location.

Address operator (&)

Indirection operator (*)

Part - B

1. **Discuss two-dimensional arrays.**

A two-dimensional array contains two subscripts. A two-dimensional array can be considered as a table made up of rows and columns and can be declared as x[i][j].

The general form of declaring the two-dimensional array is

storage-class datatype array[size1][size2];

Two-dimensional array can be initialized with value when declaring the array.

Example

The value set to the initialized value with refers to the individual element.

Following program illustrates the use of two-dimensional array in a C program.

Example Program

2. **How arrays can be passed to a function? Explain.**

An array can be passed to a function as a parameter. The name of the array is used as an argument to a function. The array name should not be enclosed within square brackets and the size of the array should not be specified.

When an array is passed to a function, the value of the array is not passed but only the address of the first array element is passed.

Following program illustrates the concept of passing arrays to a function.

Example Program

3. Discuss about structure in C.

A structure is a type of data structure whose individual element can differ in type. The keyword struct is used to specify a structure.

A template enclosed within curly braces describes the members of the structure which is treated as a single unit. The elements of the structure are members.

The structure definition takes the following form:

```
struct tag
{
    member 1;
    member 2;
    …
    …
    member n;
}
```

Structure definition should start with the keyword struct. No initialization is allowed within the structure. The syntax of the individual structure-type variable declaration can be defined as:

```
storage-class struct tag var1,var2,…;
```

Following example shows that how structure can be used.

Example

4. Define pointer. Explain the concept of pointer.

A pointer is a variable that holds the address of another object. For instance, if p is a pointer variable that contains the address of another variable called q, then p is said to be pointing to q.

```
int *p,q=5;
p=&q;
```

Two special operators to refer the memory location.

Address operator (&)

Indirection operator (*)

We can find the use of pointer as illustrated in the following example.

Example program

5. **Explain the concept ‘passing pointer to function’.**

When pointers are passed to a function as arguments, the data item associated with these pointer variables are altered within the function and then returned to the calling program. Any changes made to these data items within a function will be retained in the calling program. Thus, when a pointer is passed as a parameter, the corresponding data item is altered globally from within the called function.

In case of call by reference, the address of the actual arguments in the calling function are copied into formal arguments of the called function. If a variable is passed by value, then the changes to the corresponding data item will not be reflected in the calling program.

Program to illustrate the difference between call by value and call by reference.

Example program

Part - C

1. **Define array. Explain array in detail with example.**

An array is a group of same type that shares a common name. The array numbers are enclosed within square brackets.

Syntax:

storage-class datatype array-name[size];

Example:

int x[50],y[20],z;

The storage class is optional. Datatype is same for all the array values. The array elements start its value from x[0],x[1],x[2],… upto x[49]. It stores 50 integer values using the variable x.

The array need not to be specified, if the array size is not given, the compiler count the number of values as size. This method of array is called unsized array.

int even[ ]={24,5,20,12,6};

The size of the unsized array even is 5.

Example Program

2. **Discuss any five string functions available in C.**

The library file string.h is used in the C program as include header file.

strcat:

To combine or join two strings.

strcpy:

Stores the given value in the first string.

strcmp:

Compares the given two strings, if both strings are equal then it will return 0.

strrev:

Given string is resulted in reversal format.

strlen:

Finds the length of the given string.

Syntax of all functions with examples.

3. **Discuss in detail about call by value and call by reference.**

When pointers are passed to a function as arguments, the data item associated with these pointer variables are altered within the function and then returned to the calling program.

Call by value:

Any changes made to these data items within a function will not be retained in the calling program.

```
changev(int a,int b)
{
    int temp;
    temp=a;
    a=b;
    b=temp;
}
```

Call by reference:

Any changes made to these data items within a function will be retained in the calling program. Thus, when a pointer is passed as a parameter, the corresponding data item is altered globally from within the called function.

In case of call by reference, the address of the actual arguments in the calling function are copied into formal arguments of the called function. If a variable is passed by value, then the changes to the corresponding data item will not be reflected in the calling program.

```
changer(int *a,int *b)
{
    int temp;
    temp=*a;
    *a=*b;
    *b=temp;
}
```

Program to illustrate the difference between call by value and call by reference.

Example program

UNIT – V

Part - A

1. **Define File.**

   A group of related records from a file. Generally, this type of file is called a data type.

2. **Write the syntax for opening a file.**

   A file should be opened prior to reading from it or writing onto it.

   ```
   FILE *fopen(const char *filename, const char *mode);
   ```

3. **State the syntax for closing a file.**

   A file is closed when all the input/output operations are completed on it.

   ```
   int fclose(FILE *fp);
   ```

4. **What are the error handling functions in files?**

   ferror()

   feof()

   clearer()

   eof()

   perror()

5. **What is eof()?**

   This function checks for end-of-file. The syntax of this function is:

   ```
   int eof(int handle)
   ```

6. **Define class and object.**

   A class is a user defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.

   ```
   class fruit
   ```

   Object is a class type variable. Objects are also called instance of the class. Each object contains all members (variables and functions) declared in the class. A class is like a blueprint for an object.

   ```
   fruit mango,orange,banana;
   ```

7. **What is an object?**

   Object is a class type variable. Objects are also called instance of the class. Each object contains all members (variables and functions) declared in the class. A class is like a blueprint for an object.

   fruit mango,orange,banana;

<u>Part - B</u>

1. **Explain in detail about opening and closing a file.**

   A file should be opened prior to reading from it or writing onto it. The fopen function is used to open a stream or data file. The syntax of the fopen function is

```
FILE *fopen(const char *filename, const char *mode);
```

Example:

```
fp=open(“sample.txt”,“w”)
```

A file is closed when all the input/output operations are completed on it. The fclose function is used to close an opened file. The syntax of the fclose function is

```
int fclose(FILE *fp);
```

Example:

```
fclose(fp);
```

Opening and closing a file, is illustrated in the following program.

Example program

2. **Discuss the various operations on files available in C.**

   The following are the various functions that can be used on files.

getc()

Similar to getchar() function. This function is used for files instead of transferring the input data to the monitor.

putc()

Similar to putchar() function. The data from the file is displayed.

getw()

This function is used to read an integer value from the file pointed by the file pointer.

putw()

This function is used to write an integer value onto the file pointed by the file pointer.

fscanf()

This function is similar to the scanf() function, excepting that it works on files.

fprintf()

This function is similar to the printf() function, excepting that the fprintf() function works on files.

fread()

This function reads data from stream or file of any data type using binary representation.

fwrite()

This function writes to a stream or specified file.

Examples for all the functions.

3. Discuss about object oriented programming.

- Class
- Object
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism
- Dynamic binding
- Message passing

Part - C

1. **Explain in detail about Object Oriented Programming concept.**
    - Class
    - Object
    - Abstraction
    - Encapsulation
    - Inheritance
    - Polymorphism
    - Dynamic binding
    - Message passing