CMR
INSTITUTE OF
TECHNOLOGY

Internal Assessment Test 1 – March 2017-SCHEME & SOLUTION

Sub:	Programming in C and Data Structures							**Code:**	15PCD23
Date**:**	30/ 03/2017	Duration:	90mins	Max Marks:	50	**Sem:**	II	**Branch:**	CSE/ISE/MECH

Note: Answer any five questions:

		Marks	CO	RBT
1	**Explain structure of C program in detail. Write a program to calculate simple interest.**	**10M**	CO 2	L1, L3

- The basic structure of a C program is shown below Documentation Section

```
Documentation Section
 Link Section
 Definition Section
 Global Declaration Section
 main() Function Section
{
        Declaration Part Executable Part
}
Subprogram section
Function 1
Function 2
    .
    .
Function n
```

- The documentation section consists of a set of comment lines giving the name of the program, the name of the author and other details which the programmer would like to use later.
- The link section provides instructions to the compiler to link functions from the system library.
- The definition section contains all symbolic constants.
- There are some variables that are used in more than one function. Such variables are called global variables and are declared in the global declaration section.
- Every C program must have one main() function section.
- The subprogram section contains all the user-defined functions that are called in the main function. The main function is very important compared to other sections.

// C Program to calculate Simple Interest

```
#include<stdio.h>
int main()
{
        float p,t,r,si;
        printf("\n Enter the principal, time and rate of interest:");
        scanf("%f%f%f",&p,&t,&r);
        si=(p*t*r)/100;
        printf("\n The simple interest is %f",si);
        return 0;
}
```

2	**a) Define an operator. Explain with example.** An operator is a symbol that tells the compiler to perform specific mathematical or logical functions. C language is rich in built-in operators and provides the following types of operators − • Arithmetic Operators +,-,/,*,% Ex: a+b, x/y • Relational Operators >,<,<=,>=,==,!= Ex: a=10,b=5 a>=b, a==6 • Logical Operators &&, ||, ! Ex: a&&b • Bitwise Operators &,|,~,^,>>,<< Ex: a=3, a>>1 • Assignment Operators =, +=, -=,*=,/= Ex: i=10, i+=2	**6M**	CO 2	L1, L2
	b) Describe data types in C. Explain the syntax for variable declaration and initialization The basic data types in C are: 1.int – 2 or 4 byte 2.float – 4 bytes 3.char -1 byte 4.void Syntax for variable declaration: <datatype> variablename; Syntax for variable initialization variablename=value;	**4M**	CO 2	L1, L2
3	**a) Define an identifier. List the rules to construct an identifier. Give examples of Valid and Invalid identifier.** An identifier is a string of alphanumeric characters that begins with an alphabetic character or an underscore character that are used to represent various programming elements such as variables, functions, arrays, structures, unions and so on. • Identifier names may consist of letters, digits, and the underscore(_) character, subject to the rules given below: 1. The identifier must always begin with a letter. Some systems permit underscore as the first character. 2. ANSI standard recognizes a length of 31 characters. However, the length should not be normally more than eight characters. 3. Uppercase and lowercase are significant. That is ,the identifier Rate is not the same as rate or TOTAL. 4. The identifier name should not be a keyword.	**4M**	CO 2	L1, L2

5. White space is not allowed.

Valid Identifier: a, abc, x_y, d_123
Invalid Identifier : a bc, 1a, a$r

b) Explain different conditional decision making statements with syntax.

6M CO 2 L2

The main decision statement in C is if statement. There are four variants:

- if
- if-else
- nested if-else
- cascaded if-else or else-if ladder

If Statement (1M)

It is one-way selection statement. It is used only when there is one alternative. The syntax of if statement is:

```
if ( expression )
{
      statement-block-1;
}
statement-block-2;
```

The expression is evaluated to true or false.

- If the expression is evaluated to true, then statement-block-1 is executed and the control comes outside of if statement and the execution of further statements (statement-block-2) continues if any.
- If the expression is evaluated to be false, then statement-block 1 is skipped.

If-else Statement (1M)

It is two-way selection statement. It is used when we have to choose between two alternatives.

The syntax of if else statement is:

```
if(expression)
{
statement-block-1;
}
else
{
statement-block-2;
}
statement-block-3;
```

The expression is evaluated to true or false.

If the expression is evaluated to true, then statement-block-1 is executed and the control comes outside of if-else and statement-block-3 is executed.

If the expression is evaluated to false, then statement-block-2 is executed and the control comes outside of if-else and statement-block-3 is executed.

Nested if-else Statement (2M)

It is multi-way selection statement. It is used when an action has to be performed based on many decisions.

An if-else statement within another if-else statement: is called nested if-else statement.

The syntax of nested if-else statement is:

```
if  (  expression-1  )
{
        if  (  expression-2   )
                statement-block-1;
        else
                statement-block-2;

else
{
        if  (  expression-3   )
                statement-block-3;
        else
                statement-block-4;
}
statement-block-5;
```

The expression-1 is evaluated to true or false.

If expression-1 is evaluated to true, then expression2 is evaluated to true or false. If expression-2 is evaluated to true, then statement-block-1 is executed. If expression-2 is evaluated to false, then statement-block-2 is executed.

If expression-1 is evaluated to false, then expression-3 is evaluated to true or false. If expression-3 is evaluated to true, then statement-block-3 is executed. If expression-3 is evaluated to false, then statement-block-4 is executed.

Cascaded if-else or else if ladder Statement (2M)

It is multi-way selection statement. It is used when we must alternatives.

The syntax of cascade if-else or else if ladder statement is:

```
if  ( expression-1  )
{
        statement-block1;
else if      (expression-2  )
{
        statement-block-2;
}
else if      (expression-3)
{
        statement-block-3;
}
else   if   (expression4)
```

```
{
        statement-block4;
}
else
{
        statement-block-5;

}
statement-block6;
```

The expression is evaluated in top to bottom order. If an expression is evaluated to true, then the statement-block associated with that expression is executed and the control comes out of the entire else if ladder and continues execution from statement-block-6 if any.

If all the expressions are evaluated to false, then the last statement-block-5 (default) is executed and the control comes out of the entire else if ladder and continues execution from statement-block-6 if any.

4 a) Write a program to find greatest of the three numbers using nested if-else. **5M** CO 3 L3

```
#include <stdio.h>
int main()
{
    int n1, n2, n3;
    printf("Enter three numbers: ");
    scanf("%d %d%d", &n1, &n2, &n3);
    if (n1>=n2)
    {
        if(n1>=n3)
            printf("%d is the largest number.", n1);
        else
            printf("%d is the largest number.", n3);
    }
    else
    {
        if(n2>=n3)
            printf("%d is the largest number.", n2);
        else
            printf("%d is the largest number.",n3);
    }
        return 0;
}
```

b) Write a program to find whether a given year is leap year or not. (Consider century years) **5M**

```
#include <stdio.h>

int main()
{
  int year;
  printf("Enter a year to check if it is a leap year\n");
  scanf("%d", &year);

  if ( year%400 == 0)
    printf("%d is a leap year.\n", year);
  else if ( year%100 == 0)
    printf("%d is not a leap year.\n", year);
  else if ( year%4 == 0 )
    printf("%d is a leap year.\n", year);
  else
    printf("%d is not a leap year.\n", year);

  return 0;
}
```

5 **a) Differentiate between while and do-while loop.**

6M CO 3 L3

WHILE	**DO-WHILE**
It is an Entry Controlled Loop.	**It is an Exit Controlled Loop**

Syntax: while(condition)
{
statements;
}
	Syntax: do
	{
	statements;
	} while(condition);

In 'while' loop the controlling condition appears at the start of the loop.	**In 'do-while' loop the controlling condition appears at the end of the loop.**
The iterations do not occur if, the condition at the first iteration, appears false.	**The iteration occurs at least once even if the condition is false at the first iteration.**

b) Write a program to check whether the given number is even or odd. (Use conditional operator)

4M

```
#include<stdio.h>
int main()
{
        int n;
        printf("\n Enter the number:");
        scanf("%d",&n);
        (n%2==0)?printf("Even number"):printf("Odd number");
        return 0;
}
```

6 **a) Evaluate the following expression:**

a=5, b=6, c=7

4M CO 1 L4

d=++a * b-c;------29
d=6*6-7=29

e=a & b;--------6

f=100%20<=20-5+100%10-20==5>=1!=20--------1

g=c-- * b+d;--------71
g=7*6+29=71

b) What will be value of x in following segments.

int a , b; float x; a=4; b=5; x=b/a; **4M**

X=1.000000

int a , b; float x; a=4; b=5; x=(float)b/a;

X=1.250000

c) What is the difference between 7,7.0,'7' and "7"? **2M**

7- integer number
7.0- Float number
'7'- Character
"7"- String

7 **Explain the syntax of switch case. Write a program to perform basic arithmetic operations like addition, subtraction, multiplication and division, using switch statement. (Error Message should be displayed if number is divided by 0)** **10M** CO 3 L2

The syntax for a **switch** statement in C programming language is as follows –

```
switch(expression)
{
    case constant-expression  :
                                  statement(s);
                                  break; /* optional */

    case constant-expression  :
                                  statement(s);
                                  break; /* optional */


                     default :  /* Optional */
                                  statement(s);
}
```

The following rules apply to a switch statement −

- The expression used in a switch statement must have an integral or enumerated type, or be of a class type in which the class has a single conversion function to an integral or enumerated type.
- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
- The constant-expression for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.
- When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.
- When a break statement is reached, the switch terminates, and the flow of

control jumps to the next line following the switch statement.

- A switch statement can have an optional default case, which must appear at the end of the switch.

```
 #include<stdio.h>
int main()
{
       int a, b;
       char ch;
   printf(“\n Enter the values of a & b:”);
   scanf(“%d%d”,&a,&b);
   printf(“\n Enter the character for add, sub, mul & div:”);
   scanf(“%c”,&ch);
   switch(ch)
   {
          case ‘+’ : printf(“Sum=%d”,a+b);
                         break;
          case ‘-’ : printf(“Diff=%d”,a-b);
                         break;
           case ‘*’ : printf(“Prod=%d”,a*b);
                         break;
           case ‘/’ : if(b==0)
                              printf(“\n Number cannot be divided by zero”);
                        else
                               printf(“Div=%d”,a/b);
                         break;
           default : printf(“\n Invalid Character”);
   }
```

8	**a) Write a program and pseudo code to check whether the given number is palindrome or not.** **Pseudocode to check whether the given number is palindrome or not.**	**5M**	CO 3	L3

```
Start
Read a number
Initialize temp=number
Loop:
        Reverse the number digit by digit
End Loop
If reverse equal to the original number
     Print palindrome
Else
     Print not palindrome
End if
Stop
```

// C Program to check whether the given number is palindrome or not.

```
#include<stdio.h>

int main()
{
        int temp, num, rev, rem;
        printf ( "Enter an integer: " );
        scanf ( "%d", &num );

        temp = num;
        rev = 0;
        printf ( "Entered integer is: %d\n", num );

        while ( temp != 0 )
        {
                rem = temp % 10;
                rev = rev * 10 + rem;
                temp = temp / 10;
        }
        printf ( "The reversed number is: %d\n", rev );

        if ( num == rev )
        printf ( "The given number %d is a PALINDROME!\n", num );
        else
        printf ( "The given number %d is NOT a palindrome.\n",
num );

        return 0;
}
```

b) Write a program to print Fibonacci series up to n terms.

5M

```
#include <stdio.h>
int main()
{
   int i, n, t1 = 0, t2 = 1, nextTerm = 0;
   printf("Enter the number of terms: ");
   scanf("%d", &n);
   printf("Fibonacci Series: ");
   for (i = 1; i <= n; ++i)
   {
      // Prints the first two terms.
      if(i == 1)
      {
         printf("%d, ", t1);
         continue;
```

```
        }
        if(i == 2)
        {
            printf("%d, ", t2);
            continue;
        }
        nextTerm = t1 + t2;
        t1 = t2;
        t2 = nextTerm;
        printf("%d, ", nextTerm);
    }
    return 0;
}
```