**CMR**
**INSTITUTE OF**
**TECHNOLOGY**

# Solution/Model Answer of IAT-1 (March 2018)
# Programming in C & data Structure – 17PCD23
# II sem – All
# Dr. P. N. Singh, Professor(CSE)

1.

**a. What is an identifier? What are rules to construct identifier? Classify following as valid/invalid identifier.**
**num1, $num1, a_2, for, 1st_paper_marks.** **06**

Ans:
Identifiers - Names of variables, function, or arrays are identifiers.
Rules to construct identifier:

➔ They may contain A-Z, a-z, and _ (underscore) (no blanks)
➔ An identifier name but must be started with an alphabet or _ (specific purposes)
➔ Maximum length of name of identifier should be 31 characters (ANSI C standards) however in newer versions length of identifier name may be more.
➔ Keywords & compiler constants should not be identifier name

Valid identifiers: num1, a_2
Invalid identifiers: $num1, for, 1st_paper_marks

**b. Explain any 4 types of operators in C.** **04**

Ans:

| | |
|---|---|
| Arithmetic Operators | + (addition) |
| | - (subtraction) |
| | * (multiplication) |
| | / (division) |
| | % (modulus) |
| Relational Operators | < (less than) |
| | <= (less than or equal to) |
| | > (greater than) |
| | >= (greater than or equal to) |
| | ==(equal to) |
| | !=(not equal to) |
| Logical Operators | && (and) |
| | \|\| (or) |
| | ! (Not) |
| Assignment Operator | =, +=, -=, */, /=, %= |

2. **What is an Algorithm? Write algorithm for the following i) To find area of triangle by given 3 sides ii) To check given number is Palindrome or not.**

Ans:

Algorithm: An algorithm is step-by-step ( in the sequence ) procedure/ formula to get the solution of a task. Number of steps must be finite. Criteria of an algorithm to reduce the complexity of the solution.

i. Algorithm to find area of a triangle by given 3 sides

```
Step 1.     Start
Step 2.     Input sides a,b & c
Step 3.     s=(a+b+c)/2
Step 4.     If s <= a OR s <= b OR s <= c then
    4.1         Print 'Wrong measurement of triangle'
Step 5.     Else
    5.1         area=sqrt(s*(s-a)*(s-b)*(s-c))
    5.2         Print area
Step 6.     Endif
Step 7.     End.
```

ii. Algorithm to check given number is palindrome or not

```
Step 1.     Start
Step 2.     Input num
Step 3.     Temp=num
Step4.      REV=0
Step 5.     While temp <> 0
    5.1         Rem=temp MOD 10
    5.2         Rev=Rev*10+Rem
    5.3         Temp=(integer)Temp/10
Step 6.     Endwhile
Step 7.     If rev = num then
    7.1         Print "Number is palindrome"
Step 8.     Else
    8.1         Print "Number is not palindrome"
Step 9.     Endif
Step10.     End.
```

3.

a. **With Syntax and example explain printf() and scanf() functions.** **06**

Ans:

printf() – Library function for formatted output:

Output data can be written on to a standard output device using the library function printf(). The printf statement provides certain features that can be effectively exploited to control the alignment and spacing of printouts on the terminals. The general form of printf statement is:

```
printf ("control string", arg1,arg2, ........., argn);
```

Control string of printf function consists of three types of item :

- Character that will be printed on the screen as they appear.
- Format specification that define the output format for display of each item.
- Escape sequence characters such as \n,\t, and \b.

scanf() – Library function for formatted input

Formatted input refers to an input data that has been arranged in a particular format. Input data can be entered into the computer from a standard input device by means of the C library function scanf. In general terms, scanf function is written as

scanf (“control string”, arg1, arg2, ........., argn);

The control string specifies the field format in which the data is to be entered and the arguments arg1,arg2.....,argn specify the address of locations where the data is stored. Actually arguments are pointers which indicate where the data items are stored in computer’s memory.

Control string contains field specification which direct the interpretation of input data. It may include:

- Field (or format) specification, consisting of the conversion character %, a data type character (or type specifier, and an optional number, specifying the field width.
- Blanks, tabs, or new lines.

- The arguments are written as variables or arrays, whose type match the corresponding character groups in the control string. Each variable name must be preceded by an ampersand (&). Array names should not begin with an ampersand.

**b. Evaluate the expression: i) 100%20<=20-5+100%10-20==5>=1!=20**
**ii) a+=b*=c-=10 (Initial values, a=5,b=10 and c=15) 04**

Ans:

(i) 100%20 <= 20-5+100%10-20 ==5>=1!=20

(Arithmetic operator has greater precedence than relational operator)

0 <= -5 == 5 >= 1 != 20

(Now associativity of relational operator is left to right)

0 == 5 >= 1 != 20

0 >= 1 != 20

0 != 20

1 Ans is 1 (True)

(ii) a+=b*=c-=10

(Associativity of assignment operator is right to left)

So, c - = 10 will be 5

b*=c will be 50

and a+=b will be 55

Finally Answer is: a = 55, b= 50 and c = 5

4.

**a. Write a C program to calculate and display area of a rectangle. 5**

Ans:

```
/* area of a rectangle */
#include <stdio.h>
int main()
{
      float length, width, area;
      printf(“Enter length & width of rectangle : “);
      scanf(“%f%f”,&length, &width);  /* input from keyboard*/
      area=length*width;
      printf(“Area = % .3 square unit\n”, area);
      return (0);
}
```

Expected output:
Enter length and width: 5 4
Area = 20.000 square unit

b. **Write a C program to calculate and display sum and average of 5 integers.**

Ans:

```
#include <stdio.h>
int main()
{
      int x,num, sum=0;
      float avg;
      for(x=1;x<=5;x++)
      {
            printf(“Enter number : ‟);
            scanf(“%d”, &num);
            sum+=num;
      }
      avg=sum/5.00;
      printf(“sum = %d average = %f\n‟, sum,avg);
      return (0);
}

/* computing sum and average of 5 integers – without loop */
#include <stdio.h>
int main()
{
      int n1,n2,n3,n4,n5,sum;
      float avg;
      printf(“Enter 5 integers : ‟);
      scanf(“%d%d%d%d%d”, &n1,&n2,&n3,&n4,&n5);
      sum=n1+n2+n3+n4+n5;
      avg=sum/5.00;
      printf(“sum = %d average = %f\n‟, sum,avg);
      return (0);
}
```

Expected output:
Enter 5 integers: 10 20 30 40 50
sum = 150   average = 30.000000

5.

a. **Explain cascaded if else (else if ladder) and nested if statements with syntax and suitable example.** **5**

Ans:

Ladder of if ....else statement: for Multiple sets of conditions

```
if (expression/condition)
{statement1; statement2; ..... }
 else if (expression/condition)
   {statement1; statement2; ..... }
    else if (expression/condition)
```

```
  {statement1; statement2; ..... }
   ........
   else
    {statement1; statement2; ..... }
```

```
/* Example program to display division using if ladder*/
#include <stdio.h>
int main()
{
    int marks;
    printf(“Enter marks : “);
    scanf(“%d”,&marks);
    if(marks>=60)
         printf(“First division\n”);
   else
         if(marks>=45) /* no surplus condition is required */
         printf(“Second division\n”);
        else
              if(marks>=30)
                     printf(“Third division\n”);
              else
                     printf(“Fail\n”);
   return (0);
 }
```

Nested if ...... else statement: if within if

```
if (expression/condition)
{ if (expression/condition)
    {statement1; statement2; ..... }
   else
    {statement1; statement2; ..... }
}
else {  if (expression/condition)
      {statement1; statement2; ..... }
      else
      {statement1; statement2; ..... }
   }
```

```
/* Example program to find maximum in 3 integers using nested if*/
#include <stdio.h>
int main()
{
    int n1,n2,n3,max;
    printf(“Enter 3 numbers : “);
    scanf(“%d%d%d”,&n1,&n2,&n3);
    if(n1 > n2)
         if(n1 > n3)
              max=n1;
         else
              max=n3;
   else
         if(n2 > n3)
              max=n2;
         else
              max=n3;
   print(“Maximum = %d\n”,max);
   return (0);
}
```

**b. Explain Structure of a C Program with Example 05**

Ans:

Structure of C Program

```
/* Separate line comments –
             ------------ */
pre-processor directives /* header section */
global declarations
int main( )              /* main ( ) function section */
{
  local variables to function main ;
  statements associated with function main ;
 ………………….
  return (0);
}

udf1( )                 /* user defined functions */
{
  local variables to user defined function udf1 ;
  statements associated with udf1 ;
}
.
.
```

```
/* Example program to calculate simple interest */
#include <stdio.h>
int main( )
{
       float p,r,t,si;
       printf(“Enter principle, rate and time : “);
       scanf(“%f%f%f”, &p, &r, &t);
       si=p*r*t/100.00;
       printf(“Simple interest = %f\n”, si);
       return (0);
}
```

6. **Explain switch case statement with syntax and example of a calculator program to do simple operations like addition, subtraction, multiplication and division. 10**

```
switch statement:
switch (expression)
{
   case  condition1
           statement1;
           statement2;
           ......
           break;
    case condition2
           statement1;
           statement2;
           .......
```

```
                break;
        .....
        default:
                statement1;
                statement2;
                .......
        }
```

```
/* Arithmetic operation by switch */
int main( )
{

      float n1,n2,result;
      int choice;
      printf(“Enter 2 numbers : “);
      scanf(“%f%f”, &n1, &n2);
      printf(“1. Addition 2. Subtraction 3. Multiplication 4. Division:\n”);
      printf(“Enter your choice number : “);
      scanf(“%d”, &choice);
      switch(choice)
      {
            case 1 : result = n1+n2;
                   break;
            case 2 : result = n1-n2;
                   break;
            case 3 : result = n1*n2;
                   break;
            case 4 : if(n2 == 0)
                        { printf(“divide error!!!\n”); return (99); }
                   else
                        result = n1/n2;
                  break;
            default: printf(“Wrong choice!\n”);
      }
      printf(“Result = %f\n”, result);
      return (0);
}
```

Expected Output:

```
Enter 2 numbers : 50 25
1. Addition 2. Subtraction 3. Multiplication 4. Division:
Enter your choice number : 4
Result = 2.000000
```

7. **Write a C program to find roots of a quadratic equation of the form $ax^2+bx+c=0$, where a, b and c are coefficients. Explain the output for all possible cases.** **10**

Ans:

```
/* to find roots of quadratic equation */
#include <stdio.h>
#include <math.h>
main()
{
  float a,b,c,d,r1,r2, rpart,ipart;
  printf("Enter values of a, b and c : ");
  scanf("%f%f%f",&a,&b,&c);
  if(a==0) { printf(“The equation is not quadratic\n”); return (99); }
d=b*b-4*a*c;
if(d==0)
{
    r1=r2=-b/(2.0*a);
    printf(“Roots are real and equal: %f\n”,r1);
}
else
  if(d > 0)
   {
    r1=(b+sqrt(d))/(2*a);
    r2=(b-sqrt(d))/(2*a);
    printf("Roots are distinct : %.3f and %.3f\n",r1,r2);
   }
   else  /* computing imaginary roots*/
    {
         rpart=-b/(2.0*a);
         ipart=sqrt(-d)/(2.0*a);
         printf(“Roots are imaginary:\n”);
         printf(“ r1=%f+i*%f\n r2=%f-i*%f\n”,rpart,ipart,rpart,ipart);
    }
return (0);
}
```

8.

a. **Explain while loop with an example.** **5**

Ans:

while loop: while is a looping statement where statements are executed till the condition is satisfied/true.

Syntax:

```
while (condition)
{
          statement1;
          statement2;
          ......
}
```

Example:

```
/* while loop example to display number and message 5 times */
#include <stdio.h>
int main()
{
     int x=1;
    while(x<=5)
     {
         printf(“%d. %s\n”, x, “Kudi Punjaban Dil chura ke le gayee – Sona Sona”);
          x++;
      }
   return (0);
}
```

Expected Output:

```
1.   Kudi Punjaban Dil chura ke le gayee – Sona Sona
2.   Kudi Punjaban Dil chura ke le gayee – Sona Sona
3.   Kudi Punjaban Dil chura ke le gayee – Sona Sona
```

4. Kudi Punjaban Dil chura ke le gayee – Sona Sona
5. Kudi Punjaban Dil chura ke le gayee – Sona Sona

**b. Write a program to find whether the given year is leap or not. (consider century year) 5**

Ans:

```
/* to find whether the given year is leap year or not */
#include <stdio.h>
int main( )
{
        int year;
        printf("Enter year : ");
        scanf("%d", &year);
        if((year%4==0 && year%100 != 0)|| year%400==0)
      printf("Yes, %d is leap year.\n", year);
        else
                printf("No, %d is not leap year.\n", year);

        return (0);
}
```

Expected Output:

```
Enter year: 2016
Yes, 2016 is leap year.

Again
Enter year: 1900
No, 1900 is not leap year.

Again
Enter year: 2000
Yes, 2000 is leap year
```