JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, GUNA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course: Computer Programming Lab

Course Code: CS201

B. Tech. I Sem. (CSE, ECE, MECH, CE, CHE)

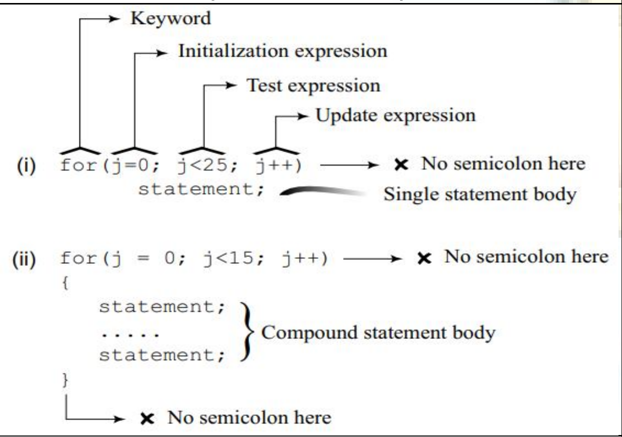
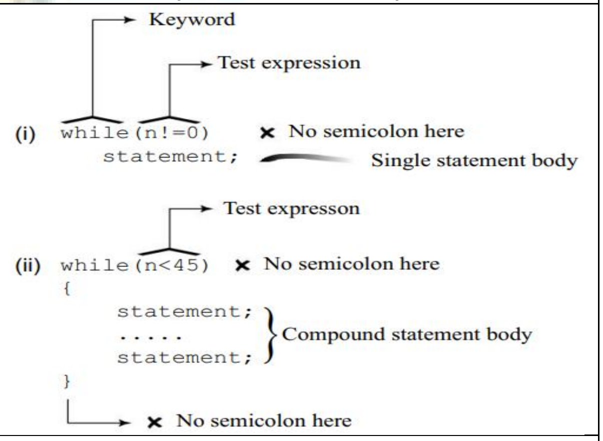
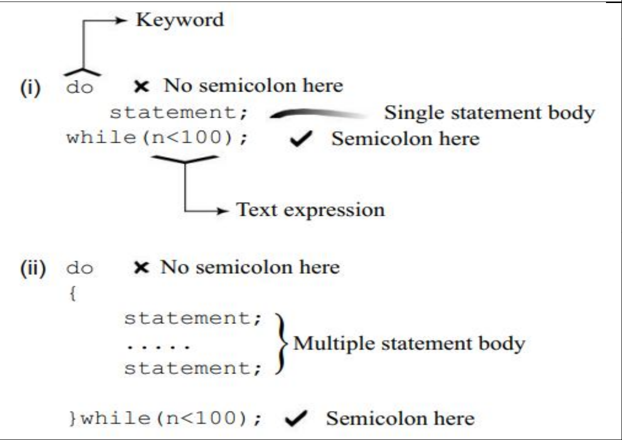
<u>Lab-9</u>

Aim: Iteration (loop) Control Statements

Iteration Control Statements: These are used to execute the sequence of statements many times until the stated condition is true. A loop in C consists of two parts, a body of a loop and a control statement. The control statement is a combination of some conditions that direct the body of the loop to execute until the specified condition becomes false. The purpose of the loop is to repeat the same code a number of times. Depending upon the position of a control statement in a program, looping statements are classified into two categories:

- **Entry controlled loop:** Condition is checked **before** executing the body of a loop. May execute **zero times** if condition fails in the first iteration. Examples: **for** and **while** loop statements
- **Exit controlled loop:** Condition is checked **after** executing the body of a loop. Executes **at least once**. Example: **do-while** loop statement

Note: goto is unconditional jump statement and works as loop if it is used with **if** to make backward jump.

Syntax for **for** loop	Syntax for **while** loop
	
Syntax for **do-while** loop	Syntax for unconditional **goto** statement
	**Forward Jump**: goto label; …….. …….. label : statement; — **Backward Jump**: label : statement; …….. …….. goto label;
	Syntax for conditional **goto** with **if** statement

	if (condition)
	goto label;
	……..
	……..
	label: statement; — label : statement;
	………
	……..
	if (condition)
	goto label;

Memorable facts about loop statements

- At most three expressions, namely **initialization, test** and **update**, are required in loop statements. Execution of these expressions takes place in a predefined order.

- In **for** and **while** loops, **initialization** expression always executes **first** and **only once**, then after **test** expression is evaluated and if it is found true, finally the **update** expression is executed and updated. In the **do-while** loop, test expression is executed in the last.

- Execution of test and update expressions repeats until the test condition false. When the test expression is false, the loop terminates.

- In all three loops statements, any of three or all three expressions can remain undefined. If none of the expression is defined in **for** loop, it behaves as an **infinite loop**. Syntax: for (; ;)

- **while** (true) and **do-while** (true) also behave as **infinite** loop statements.

- **'for'** loop is preferred in use when the exact number of iterations is already known.

- **while** and **do-while** loops are preferred in use when the exact number of iterations is unknown.

- In most of the problems, any of three loop statements can be applied.

- All three loops can also be used as **nested loops**.

- In the nested loops, inner loop always executes more number of times in comparison to outer loops.

- Any of three loop statements can be replaced by **if-goto** statement which makes conditional backward jump in the program.

- In all three loop statements, curly braces { } are optional in the case of **single** statement body and compulsory in the case of **multiple** statements body.

- In most of the repetitive addition and multiplication problems, a variable is initialized with the value of **0** and **1** for respectively.

Loop statements with 'break' and 'continue' statements

- **break:** this statement **terminates** the execution of a loop statement and control is transferred to statement immediately following the loop. The break statement breaks loops one by one, i.e., in the case of nested loops, it breaks the inner loop first and then proceeds to outer loops.

- **continue:** this statement is used to **skip** the current iteration and move on to the next iteration without executing the statements below the 'continue' in the loop body.

Example#1: To print the sum of first n integer numbers using **for** loop.

```
#include <stdio.h>
int main()
{
   int num, count;
   int sum = 0;               //variable initialization
   printf ("Enter a positive integer: ");
   scanf ("%d", &num);
   for(count = 1; count <= num; ++count)
   {
      sum += count;              //sum = sum + count;
   }
   printf ( "Sum = %d", sum);
   return 0;
}
```

Example#2: To print no of digits and sum of digits of a number using **while** loop.

```
#include <stdio.h>
int main()
{   int num, nd=0, sd=0, rem;
    printf("Enter a Number : ");
    scanf("%d", &num);
    while (num > 0)
    {
       rem = num % 10;   nd = nd + 1;
       sd = sd + rem;       num = num / 10;
     }
printf ("Number of digits = %d", nd);
printf ( "\nSum of digits = %d", sd);
 return 0;
}
```

Example#3: Program to print ASCII values (from 0 to 255) of all characters using **do-while** loop.

```
#include <stdio.h>

int main()
{
  int i = 0;
   do
   {
 printf ("ASCII value of character %c = %d\n", i, i);
      i++;            //post increment by 1 i.e. i = i+1
   } while (i <= 255);

return 0;
}
```

Example#4: To print the table of a number using **if-goto** loop.

```
#include <stdio.h>
int main()
{
int num, i = 1;
printf("Enter the number whose table to be printed ");
scanf ("%d", &num);
table:            //label for conditional backward jump
printf ("%d x %d = %d\n", num, i, num*i );
i++;
 if ( i <=10 )                    //condition
 goto table;                    //conditional goto statement
return 0;
}
```

Example#5: Program to add 10 numbers until the user enters a negative using **for-break** loop.

```
#include <stdio.h>
int main() {
  int i;
  double number, sum = 0.0;
  for (i = 1; i <= 10; ++i)
   {
    printf ("Enter n %d: ", i);
    scanf ("%lf", &number);
        if (number < 0.0)
        break;
    sum += number;        //sum = sum + number;
    }
  printf("Sum = %.2lf", sum);
  return 0;
}
```

Example#6: To print sum of 10 positive numbers (skip negative numbers) using **for-continue** loop.

```
#include <stdio.h>
int main() {
  int i;
  double number, sum = 0.0;
  for (i = 1; i <= 10; ++i)
    {
    printf ("Enter a n %d: ", i);
    scanf ( "%lf", &number);
      if (number < 0.0)
      continue;
    sum += number;          // sum = sum + number;
    }
  printf("Sum = %.2lf", sum);
  return 0;
}
```

For first turn of the week:

Exercise#1: Write **user input** C program using **for** and **if-goto** loop statements to calculate the factorial of a number and display the result. Write the algorithm and draw flowchart in your practical book.

Exercise#2: Write **user input** C program using **while** loop to find whether the input number is **Armstrong** number or not. Write the algorithm and draw flowchart in your practical book.
(**Armstrong Number:** A positive integer number that is equal to the sum of each digit of the number powered to its total digits count. For example $153 = 1^3 + 5^3 + 3^3$; $1634 = 1^4 + 6^4 + 3^4 + 4^4$)

Exercise#3: Write **user input** C program using **do-while** loop statement to calculate the greatest common divisor (GCD) and highest common factor (HCF) of two number**s**. Write the algorithm and draw flowchart in your practical book.

For second turn of the week:

Exercise#4: Write **user input** C program using a **suitable loop** statement to find the total count of digits and frequency of each digit in a given integer number. For example: integer number "5474274" has total count of digits 7 and frequency of digits 5, 4, 7 and 2 is 1, 3, 2 and 1 respectively.

Exercise#5: Write **user input** C program using a **suitable loop** statement which accepts angle $(\boldsymbol{\theta})$ and initial velocity $(v_0\ in\ m/sec)$ of a projectile as inputs to display the maximum vertical height $\left(\frac{v_0^2\ Sin^2\boldsymbol{\theta}}{2g}\right)$, maximum horizontal range $\left(\frac{v_0^2\ Sin\ 2\boldsymbol{\theta}}{g}\right)$, and time of flight $\left(\frac{2v_0 Sin\ \boldsymbol{\theta}}{g}\right)$ in tabular form as shown below. Verify that the projectile achieves maximum range at the angle of 45^o ($\pi/4\ radian$). Value of gravitational force $g = 9.8\ m/sec^2$.

Angle in degree	Maximum Height	Maximum Range	Time of Flight
5			
10			
:			
90			

Exercise#6: Write **user input** C program using **suitable nested loop** statement that accepts name, enroll number, and positive integer marks scored in three subjects of N students of a class as an input and prints the student result '**Pass'** if marks in each subject are >= 35 and average marks is >= 40 otherwise **'Fail'**. Display the mark sheet in the format as shown below.

Name	Enroll. No.	Sub1	Sub2	Sub3	Average Marks	Result

<u>Practice Questions</u>

(No need to include in your Practical Book)

1. Write a program to read a number N and to print all its divisors (except 1 and itself) otherwise print number is prime.

2. Write a program to find that given number is **palindrome** or not. (**Palindrome Number:** A number that remains the same when its digits are reversed. For example 12321)

3. Write a program to find that given number is **perfect** number or not. (**Perfect Number:** A positive integer number that is equal to the sum of its proper divisors excluding itself. For example 6, 28, 496 etc)

4. Write a C program to check whether a number is **strong** number or not. (**Strong** N**umber:** A positive integer number that is equal to the sum factorial of its digits. For example 145 =1! + 4! + 5!)

5. Write a C program to swap first and last digits of an integer number.

6. Two numbers are entered through the keyboard. Write a program to find the value of one number raised to the power of another. (For example, x and y are input variables and output is x^y)

7. Write a program to enter the numbers till the user wants and at the end it should display the count of positive, negative and zeros entered.

8. Write a program to create a list of the leap years found in a given range of years.

9. Write a C program to convert Binary to Octal number system.

10. Write a C program to convert Decimal to Binary number system.