Name: Shreyans Tatiya
Batch: C5_3 Roll No.: 16010123325

Experiment / assignment / tutorial No. 1

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

TITLE: Write a program for:
a. Program to find area and circumference of various Geometric shapes.
b. Program to calculate EMI (Equated Monthly Instalment) of loan amount if principal, rate of interest and time in years is given by the user.
$(E = (P.r.(1+r)^n) / ((1+r)^n – 1)$

AIM: Write a program for:
a. Program to find area and circumference of various Geometric shapes.
b. Program to calculate EMI (Equated Monthly Instalment) of loan amount if principal, rate of interest and time in years is given by the user.
$E = (P.r.(1+r)^n) / ((1+r)^n – 1)$

Expected OUTCOME of Experiment:

1. Find area and circumference of various Geometric shapes
2. To calculate EMI

Books/ Journals/ Websites referred:

1. Programming in ANSI C, E. Balagurusamy, 7 th Edition, 2016, McGraw-Hill Education, India.
2. Structured Programming Approach, Pradeep Dey and Manas Ghosh, 1 st Edition, 2016, Oxford University Press, India.
3. Let Us C, Yashwant Kanetkar, 15th Edition, 2016, BPB Publications, India.

Problem Definition:

Problem 1: Area and Circumference of any shape(will be given by instructor) (example Circle) Ask the user to enter the value of the radius of a circle. Put the values in the formula for finding area of a circle and circumference of a circle and print the outcome for area of a circle and circumference of a circle

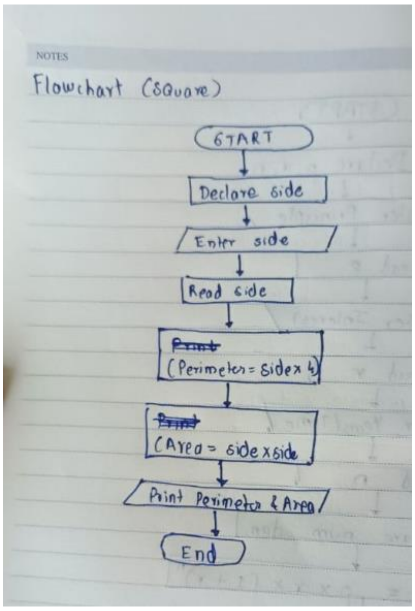
Problem 2: Calculating EMI Ask the user to enter the value of principal amount, rate of interest and time (in years).Store the value in E and print the final monthly instalment E as an outcome.

Formula to be used: $(E = (P.r.(1+r)^n) / ((1+r)^n – 1)$

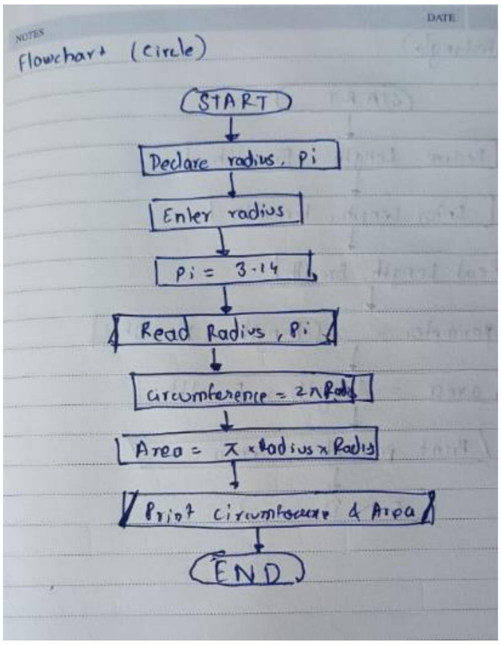
Flowchart:

1)

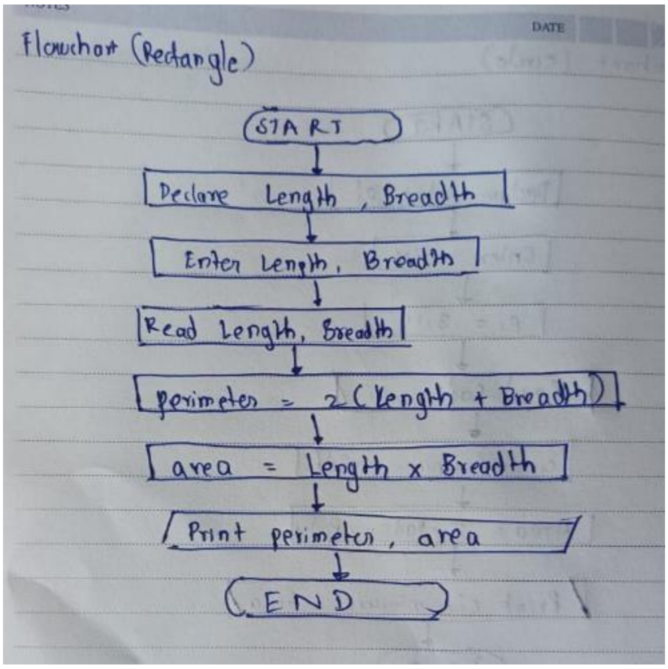
a) Square



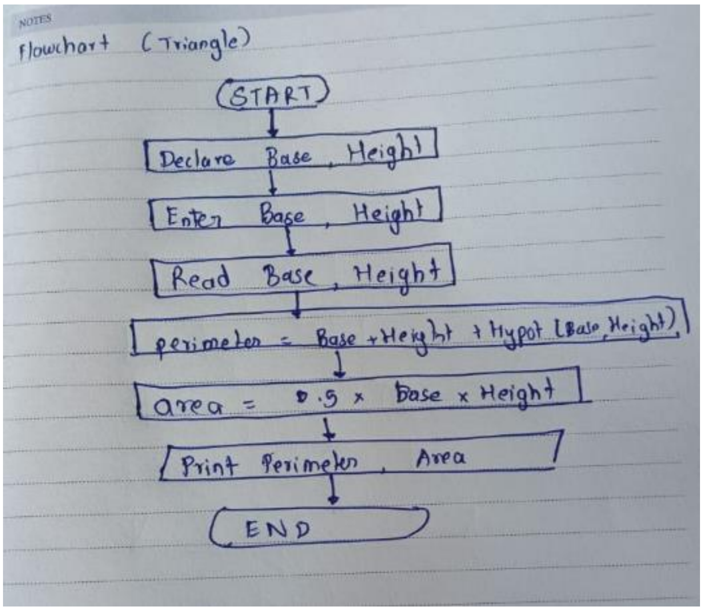
b) circle



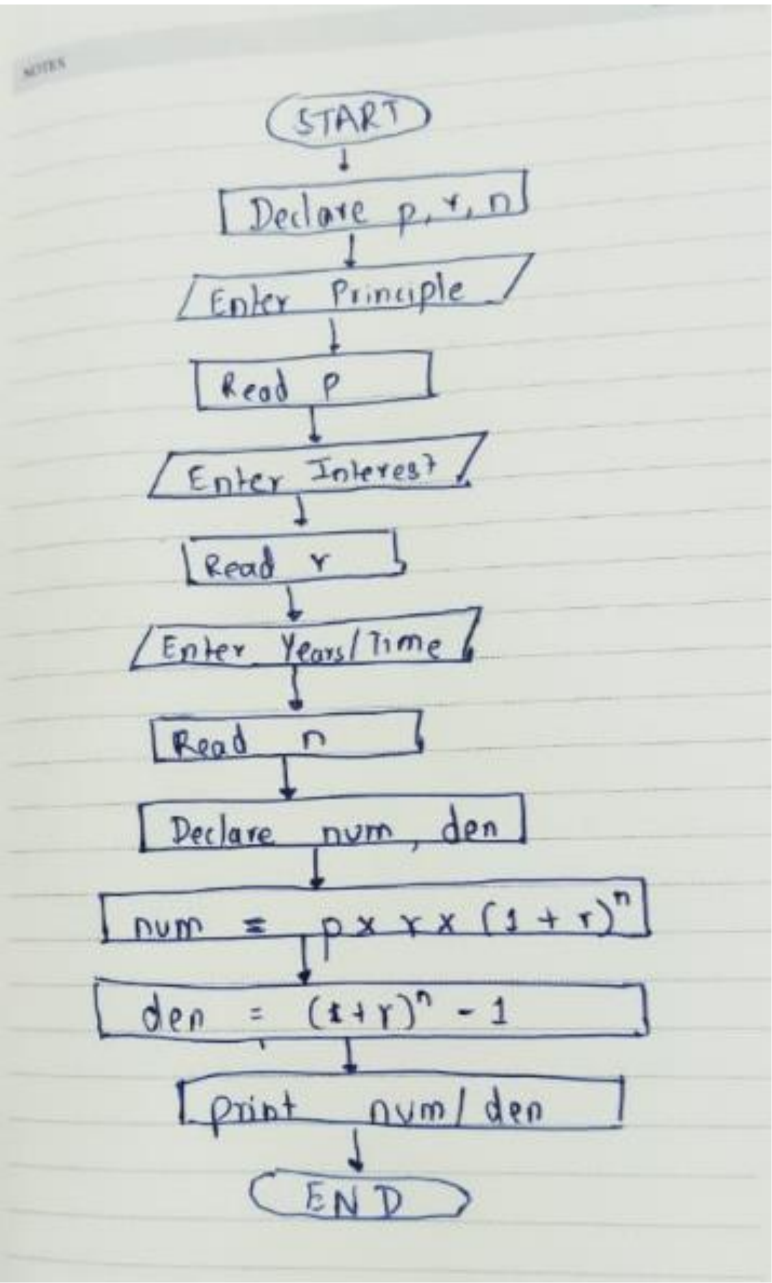
c) rectangle



d) triangle

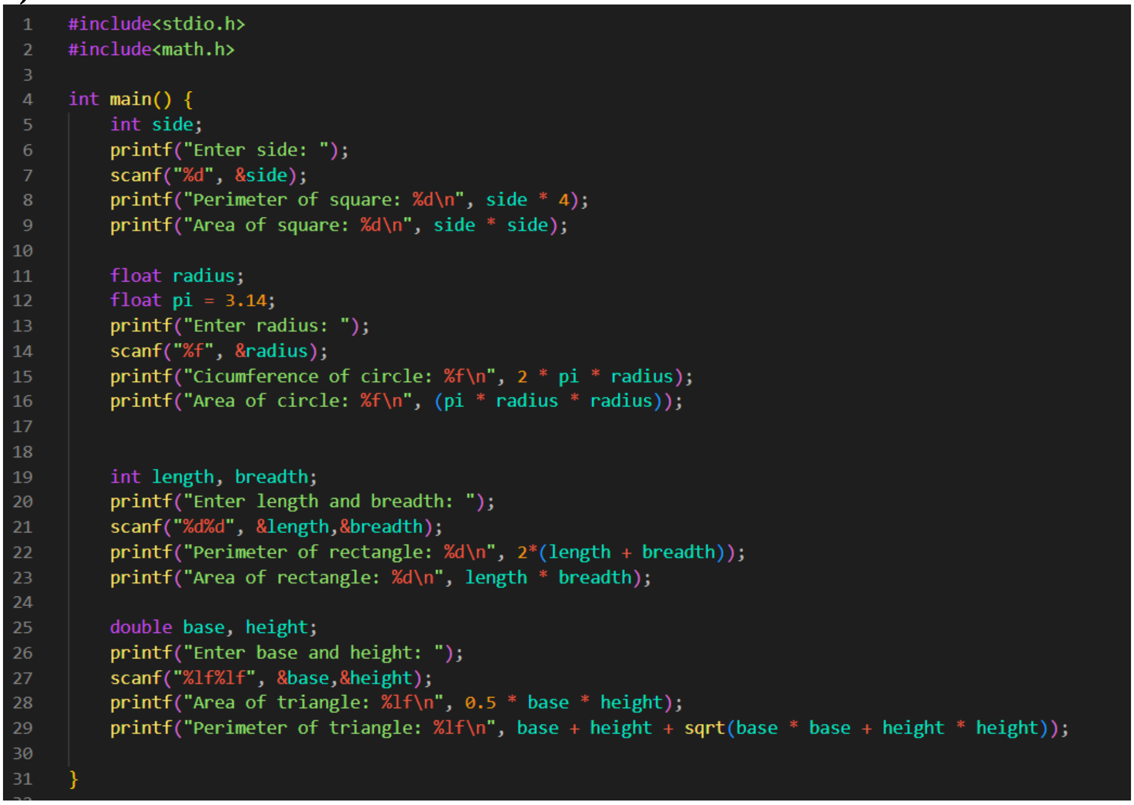


2)


NOTES
START
Declare p, r, n
Enter Principle
Read P
Enter Interest
Read r
Enter Years/Time
Read n
Declare num, den
num = p x r x (1 + r)^n
den = (1+r)^n - 1
print num/den
END

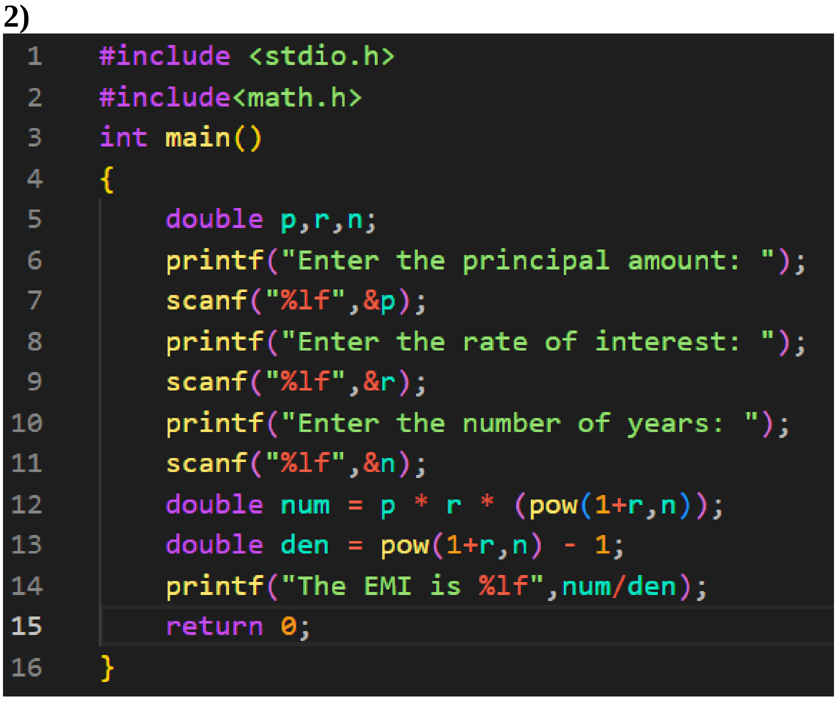
Implementation details

1)



```c
#include<stdio.h>
#include<math.h>

int main() {
    int side;
    printf("Enter side: ");
    scanf("%d", &side);
    printf("Perimeter of square: %d\n", side * 4);
    printf("Area of square: %d\n", side * side);

    float radius;
    float pi = 3.14;
    printf("Enter radius: ");
    scanf("%f", &radius);
    printf("Cicumference of circle: %f\n", 2 * pi * radius);
    printf("Area of circle: %f\n", (pi * radius * radius));


    int length, breadth;
    printf("Enter length and breadth: ");
    scanf("%d%d", &length,&breadth);
    printf("Perimeter of rectangle: %d\n", 2*(length + breadth));
    printf("Area of rectangle: %d\n", length * breadth);

    double base, height;
    printf("Enter base and height: ");
    scanf("%lf%lf", &base,&height);
    printf("Area of triangle: %lf\n", 0.5 * base * height);
    printf("Perimeter of triangle: %lf\n", base + height + sqrt(base * base + height * height));

}
```

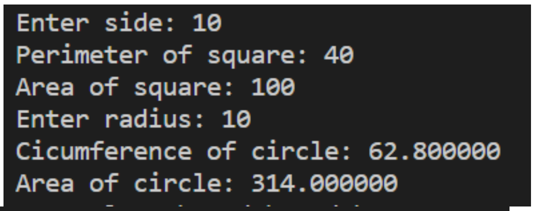
2)



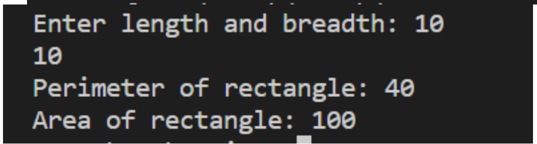
```c
#include <stdio.h>
#include<math.h>
int main()
{
    double p,r,n;
    printf("Enter the principal amount: ");
    scanf("%lf",&p);
    printf("Enter the rate of interest: ");
    scanf("%lf",&r);
    printf("Enter the number of years: ");
    scanf("%lf",&n);
    double num = p * r * (pow(1+r,n));
    double den = pow(1+r,n) - 1;
    printf("The EMI is %lf",num/den);
    return 0;
}
```

Output(s):

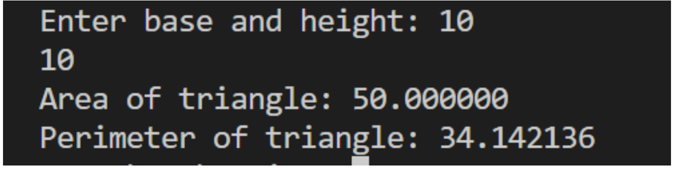
1)



```
Enter side: 10
Perimeter of square: 40
Area of square: 100
Enter radius: 10
Cicumference of circle: 62.800000
Area of circle: 314.000000
```

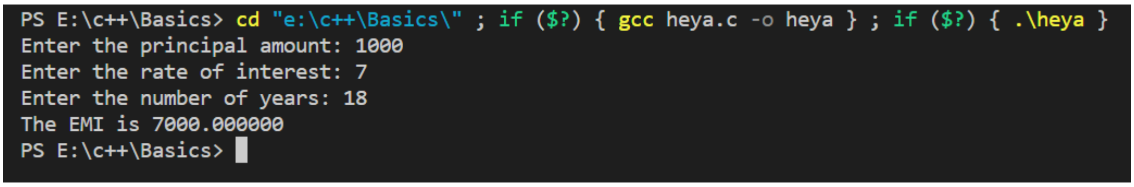


```
Enter length and breadth: 10
10
Perimeter of rectangle: 40
Area of rectangle: 100
```



```
Enter base and height: 10
10
Area of triangle: 50.000000
Perimeter of triangle: 34.142136
```

2)



```
PS E:\c++\Basics> cd "e:\c++\Basics\" ; if ($?) { gcc heya.c -o heya } ; if ($?) { .\heya }
Enter the principal amount: 1000
Enter the rate of interest: 7
Enter the number of years: 18
The EMI is 7000.000000
PS E:\c++\Basics> 
```

Conclusion:

Successfully executed out Experiment 1.

Post Lab Descriptive Questions

1. **What are the basic data types in C?**
2. **Write a table for Operator Precedence and Associativity.**

Answers:

1)

In C programming language, there are three categories of data types: primitive, user-defined, and derived data types.

Primitive data types are the most basic data types that are used for representing simple values such as integers, float, characters, etc

The following table lists the most commonly used primitive data types in C, along with their size, range, and format specifier:

Table

Data Type	**Size (bytes)**	**Range**	**Format Specifier**
short int	2	-32,768 to 32,767	%hd
unsigned short int	2	0 to 65,535	%hu
unsigned int	4	0 to 4,294,967,295	%u
int	4	-2,147,483,648 to 2,147,483,647	%d
long int	4	-2,147,483,648 to 2,147,483,647	%ld
unsigned long int	4	0 to 4,294,967,295	%lu
long long int	8	-(2^63) to (2^63)-1	%lld
unsigned long long int	8	0 to 18,446,744,073,709,551,615	%llu
signed char	1	-128 to 127	%c

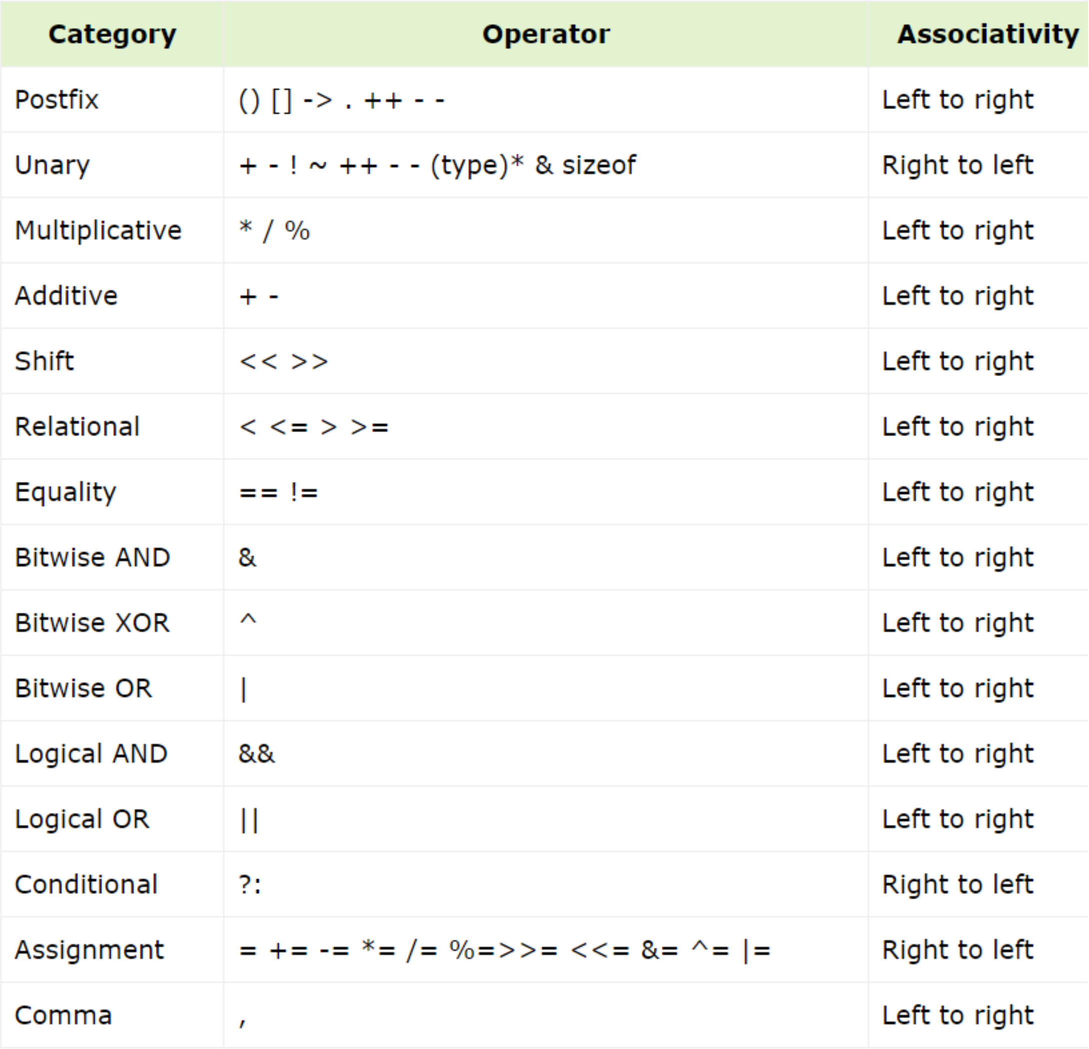
Data Type	Size (bytes)	Range	Format Specifier
unsigned char	1	0 to 255	%c
float	4	1.2E-38 to 3.4E+38	%f
double	8	1.7E-308 to 1.7E+308	%lf
long double	16	3.4E-4932 to 1.1E+4932	%Lf

User-defines data types are defined by the user and can be of any type, including primitive and derived data types.

Derived data types are data types that are derived from the primitive or built-in data types. Examples of derived data types include arrays, pointers, structures and union

2)

Here, operators with the highest precedence appear at the top of the table, those with the lowest appear at the bottom. Within an expression, higher precedence operators will be evaluated first.



Category	Operator	Associativity
Postfix	() [] -> . ++ - -	Left to right
Unary	+ - ! ~ ++ - - (type)* & sizeof	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Shift	<< >>	Left to right
Relational	< <= > >=	Left to right
Equality	== !=	Left to right
Bitwise AND	&	Left to right
Bitwise XOR	^	Left to right
Bitwise OR	|	Left to right
Logical AND	&&	Left to right
Logical OR	||	Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %=>>= <<= &= ^= |=	Right to left
Comma	,	Left to right

Date: _____________ **Signature of faculty in-charge**