B.Tech I Year

Regular Course Handbook

Subject Name: Programming for Problem Solving (Unit-2)

BCS101 / BCS201: PROGRAMMING FOR PROBLEM SOLVING

Content	Contact Hours
Unit -1:	8
Introduction to Components of a Computer System: Memory, Processor, I/O Devices, Storage, Operating System, Concept of Assembler, Compiler, Interpreter, Loader and Linker. **Idea of Algorithm:** Representation of Algorithm, Flowchart, Pseudo Code with Examples, From Algorithms to Programs, Source Code. **Programming Basics:** Structure of C Program, Writing and Executing the First C Program, Syntax and Logical Errors in Compilation, Object and Executable Code. Components of C Language. Standard I/O in C , Fundamental Data types, Variables and Memory Locations, Storage Classes.	
Unit-2:	8
Arithmetic Expressions and Precedence : Operators and Expression Using Numeric and Relational Operators, Mixed Operands, Type Conversion, Logical Operators, Bit Operations, Assignment Operator, Operator precedence and Associatively. **Conditional Branching:** Applying if and Switch Statements, Nesting if and Else and Switch.	
Unit-3:	8
Iteration and Loops: Use of While, do While and for Loops, Multiple Loop Variables, Use of Break , Goto and Continue Statements. **Arrays:** Array Notation and Representation, Manipulating Array Elements, using Multi -Dimensional Arrays. Character Arrays and Strings, Structure, union, Enumerated Data types, Array of Structures, Passing Arrays to Functions.	
Unit-4:	8
Functions: Introduction, Types of Functions, Functions with Array, Passing Parameters to Functions, Call by Value, Call by Reference, Recursive Functions. **Basic of searching and Sorting Algorithms:** Searching & Sorting Algorithms (Linear Search , Binary search , Bubble Sort, Insertion and Selection Sort)	
Unit-5:	8
Pointers: Introduction, Declaration, Applications, Introduction to Dynamic Memory Allocation (Malloc, Calloc, Realloc, Free), String and String functions , Use of Pointers in Self-Referential Structures, Notion of Linked List (No Implementation) **File Handling:** File I/O Functions, Standard C Preprocessors, Defining and Calling Macros and Command-Line Arguments.	

Course Outcome:

Course Outcome (CO)		Bloom's Level
At the End of Course , the Student will be Able to Understand		
CO 1	To Develop Simple Algorithms for Arithmetic and Logical Problems.	K_2, K_3
CO 2	To Translate the Algorithms to Programs & Execution (In C Language).	K_3
CO 3	To Implement Conditional Branching, Iteration and Recursion.	K_3
CO 4	To Decompose a Problem into Functions and Synthesize a Complete Program Using Divide and Conquer Approach.	K_4
CO 5	To Use Arrays, Pointers and Structures to Develop Algorithms and Programs.	K_2, K_3

K_1- Remember, K_2- Understand, K_3- Apply, K_4- Analyze , K_5- Evaluate , K_6- Create

Text Books:

1. Schaum's Outline of Programming with C by Byron Gottfried , McGraw-Hill
2. The C programming by Kernighan Brain W. and Ritchie Dennis M., Pearson Education .
3. Computer Basics and C Programming by V.Rajaraman , PHI Learning Pvt. Limited, 2015.
4. Computer Concepts and Programming in C, E Balaguruswami, McGraw Hill
5. Computer Science- A Structured Programming Approach Using C, by Behrouz A. Forouzan, Richard F. Gilberg, Thomson, Third Edition , Cengage Learning - 2007.
6. Let Us C By Yashwant P. Kanetkar.
7. Problem Solving and Program Design in C, by Jeri R. Hanly, Elliot B. Koffman, Pearson Addison-Wesley, 2006.
8. Programming in C by Kochan Stephen G. Pearson Education – 2015.
9. Computer Concepts and Programming in C by D.S. Yadav and Rajeev Khanna, New Age International Publication.
10. Computer Concepts and Programming by Anami, Angadi and Manvi, PHI Publication
11. Computer Concepts and Programming in C by Vikas Gupta, Wiley India Publication
12. Computer Fundamentals and Programming in C. Reema Thareja, Oxford Publication

B.Tech First Year: Regular Course Lecture Plan Session 2022-23

Subject Name	Programming for Problem Solving

Unit No.	Unit Name	Syllabus Topics	Lecture No
1	Introduction to Programming: Introduction to components of a computer system:	Memory, processor, I/O Devices, storage	1
		Operating system and it's type, Introduction to low level & high level languages	2
		IDE role for development , Concept of assembler, compiler, Interpreter, loader and linker	3
	Idea of Algorithm:	Representation of Algorithm, Flowchart	4
		Pseudo code with examples	5
	Programming Basics:	Structure of C program: writing and executing the first C program, types of errors	6
		Components of C language: Standard I/O in C	7
		Fundamental data types, Variables and memory locations	8
2	Arithmetic expressions & Conditional Branching: Arithmetic expressions and precedence:	Operators and expression using numeric and relational operators, mixed operands	9
		Logical operators, bit operations, assignment operator	10
		Operator precedence and associatively, Implicit and explicit type conversion	11
	Conditional Branching:	Applying if	12
		Nesting if else	13
		Else if ladder	14
		Switch statements, use of break and default with switch.	15
3	Loops & Functions: Iteration and loops:	Use of while and for loops	16
		Concept of do while loop, break and continue	17
		Nested Loop and multiple loop variables	18
	Functions:	Introduction to function and types of functions	19
		Call by value and call by reference	20
		Storage Class	21
		Recursive functions	22

B.Tech First Year: Regular Course Lecture Plan Session 2022-23

Subject Name	Programming for Problem Solving

Unit No.	Unit Name	Syllabus Topics	Lecture No
4	Arrays & Basic Algorithms: Arrays:	Array notation and representation, manipulating array elements	23
		Using multi dimensional arrays	24
		Functions with array Passing arrays to functions	25
		Character arrays and strings	26
		Structure & Union	27
		Enumerated data types	28
	Basic Algorithms:	Searching : Linear Search	29
		Binary Search	30
		Basic Sorting Algorithms : Bubble Sort	31
		Selection Sort	32
		Insertion Sort, Notion of order of complexity	33
5	Pointer& File Handling: Pointers:	Introduction, declaration, applications of pointer	34
		Introduction to dynamic memory allocation:- malloc, calloc, realloc and free	35
		Linked List: Use of pointers in self-referential structures notion of linked list (no implementation)	36
	File handling:	File I/O functions	37
		File Programs	38
		Standard C preprocessors, Types of Preprocessor Directives	39
		Command-line arguments	40

Signature	
Name of Subject Head	Ms. Radhika Jindal

Unit :- 2

V.imp-

Ques What is an operator ? List all the operators used in C. Give example. (2015-16, 2018-19)

Ans Operators are special symbols whose meaning is already known to C compiler. There are 45 operators in C classified as :-

① **Unary operators** :- Operators that need only one operating value or operand to complete its task is termed as unary operator.

Ex :- (!) logical not (~) complement

② **Binary operators** :- operators that need two operand to complete its task termed as binary operator.

Ex :- + (Addition) , * (multiplication)

③ **Ternary operators** :- operators that need three operand to perform its task is termed as conditional operator.

Ex :- exp1 ? exp2 : exp3

Note :- It first evaluate the exp1 condition, if it is true then exp2 is evaluated, if the condition is false then exp3 is evaluated.

* The operators are classified eight (8) general categories

① **Arithmetic operators** :- This operators which helps us to carryout basic arithmetic operations.

Ex :- addition, subtraction, multiplication, division.

operator	Meaning	Example
+	Addition	1 + 2 = 3
-	Substraction	3 - 2 = 1
*	Multiplication	2 * 2 = 4
/	Division	2/2 = 1
%	Modulo Division	10 % 3 = 1

(2) Relational operators:- The operator which are used to form conditions for comparing two operands or values are termed as relational operators. There are six relational operators

Operator	Meaning	Example	Return value
<	is less than	3<5	1
<=	is less than or equal to	4<=2	0
>	is greater than	7>5	1
>=	is greater than or equal to	3>=5	0
==	equal to	6==6	1
!=	not equal to	5!=5	0

(3) Logical operators: The operators which are used to combine the results of two or more conditions are termed as logical operator. There are three logical operators used in 'C'.

Operator	meaning	Example	Return value
&&	Logical AND	(9>2)&&(6>4)	1
||	Logical OR	(9>2)||(3>4)	1
!	Logical Not	!4	0

Truth Table:

AND (&&)

T	T	T
T	F	F
F	T	F
F	F	F

OR (||)

T	T	T
T	F	T
F	T	T
F	F	F

④ Assignment operator:- The operators which are used to assign the right hand side computed value to the left hand side variable is termed as assignment operator.

Syntax:- identifier = expression ; like int r = 2, ac ;

ac = 3.14 * r * r

⑤ Increment / Decrement operator:- The operator which is used to increment or decrement the value of variable by one is termed as increment / decrement operator.

Ex:- ++ , --

Pre - Inc/Dec operator	Post-Inc/Dec operator
① Operator comes before the operand ② Value is increment first & then it is assigned int n = 2 , y ; y = ++n Then n = 3 , y = 3	operator comes after the operand. Value is assign first & then it is incremented. int n = 2, y ; y = n++ Then n = 3, y = 2

(6) Bitwise operator :- The operators which are used to perform operation at bit level are termed as bitwise operator. There are six bitwise operators used in 'C'.

Operator	Meaning	Example	Return value
&	Bitwise AND	5&7	5
|	Bitwise OR	5|7	7
^	Bitwise XOR	5^7	2
~	Complement	~5	-6
<<	Left shift	4<<2	16
>>	Right shift	16>>1	8

(7) Special operators :- The operator like comma, size of are termed as special operator.

- Comma operator :- It is used to seprate multiple values in an expression or a statement.

  Like int i=2, j;

  j = i+ (1, 2, 3, 4, 5);

  j = 7

- Size of operator :- It is used to find the number of bytes occupied by a data type variable or a value.

  Like int i;

  sizeof (int) = 2

  size of (i) = 2

  sizeof (5) = 2

Que: 2 What are the different type of relational operator? Discuss each with example ? (2014-15, 2017-18, 2019-20)

'or'

what is an assignment statement ? Give the general format of an assignment statement (2014-15, 2019-20)

'or'

Explain the following operators in c language (2016-17)

(i) Relational

(ii) Logical

(iii) Conditional

'or'

What is an operator? Explain the arithmetic, relational, logical and assignment operator in C language. (2015-16) (2019-20, 2020-21)

'or'

what are bitwise operator ? How they are different from logical operators. (2017-18)

'or'

What are conditional operator ? Explain with suitable example. (2018-19)

'or'

Differentiate post increment and pre increment with example. (2016-17)

Ans Prefer Que-1 answer for all operators detail.

Imp. Examples of conditional operator:-

```
#include<stdio.h>
int main()
{
```

```
int a=5, b;                    // variable declaration
b=((a==5)?(3):(2));            // conditional operator
printf("The value of 'b' variable is : %d", b);
return 0;
```

Explanation: In this code we have declared two variable i.e 'a' and 'b' and assign 5 value to the 'a' variable. After the declaration, we are assigning value to the 'b' variable by using the conditional operator. If the value of 'a' is equal to 5 then 'b' is assigned with a 3 value otherwise 2.

O/P :- The value of 'b' variable is : 3

Example of one relational operator

```
#include <stdio.h>
int main()
{
    int m=40, n=20;
    if (m==n)
    {
        printf("m & n are equal");
    }
    else
        printf("m and n are not equal");
}
```

O/P :- m and n are not equal.

Q3 Write a C program to find the largest of three number using ternary operator. (2016-17)

Sol^n

```
#include <stdio.h>
int main()
{
  int n1 = 5, n2 = 10, n3 = 15, max;
  max = (n1 > n2) ? (n1 > n3 ? n1 : n3) :
                    (n2 > n3 ? n2 : n3);
  printf("Largest number among %d, %d and
          %d is %d.", n1, n2, n3, max);
  return 0;
}
```

O/P:- Largest number among 5, 10 and 15 is 15.

Imp

Q4 Differentiate between precedence and associativity (2016-17)

or,

Define (i) variable (ii) constant (iii) Associativity (iv) Precedence (2017-18, 2019-20, 2020-21)

Ans Precedence & associativity of operator:

① Precedence is a term which describe the order of execution of operators in an expression having different priority. The highest precedence operator is applied first, followed by the next highest and so on.

Ex:- * has high precedence than +

Note:- For variable & constant read c tokens in unit 1st.

② Associativity is a term which describe the order of execution of operators in an expression having same priority. It tell how the operator of same precedence are grouped and how the expression will be evaluated.

operator	Meaning of operator	Associativity	Priority
() [] -> .	Function call Array element reference Indirect member selection Direct member selection	Left to right	1
! ~ + - ++ -- & * sizeof (type)	Logical Negation Bitwise (1's) complement Unary plus Unary minus Increment Decrement Operator (Address) Pointer reference Returns the size Type cast (conversion)	Right to left	2
* / %	Multiply Divide Remainder	Left to right	3
+ -	Binary plus (addition) Binary minus (subtraction)	Left to right	4

operators	Meaning of operators	Associativity	Priority
<< >>	left shift right shift	left to right	5
< <= > >=	less than less than or equal greater than greater than or equal	left to right	6
== !=	Equal to Not equal to	left to right	7
&	Bitwise AND	left to right	8
^	Bitwise exclusive OR	left to right	9
|	Bitwise OR	left to right	10
&&	Logical AND	left to right	11
||	logical OR	left to right	12
?:	conditional operator	Right to left	13
=, *=, /=, %=, -=, &=, ^=, |=, <<=, >>=	Assignment operators	Right to left	14
,	comma operator	Left to right	15

Ques- What is expression? Evaluate the following expression.

(i) n = a - b / 3 + c * 2 - 1 when a = 9, b = 12, c = 3 (2015-16)

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

(iii) a+ = b *= c - - = 5

(iv) 9 - 12 / 6 * (2 - 1)

{ for (ii) & (iii) a = 3, b = 5, c = 8) (2016-17)

Ans (i) $n = a - b/3 + c * 2 - 1$

when $a = 9 \quad b = 12 \quad c = 3$

Put the value of a, b, c

$$n = 9 - \underline{12/3} + \underline{3 * 2} - 1$$

as *, / having higher priority than + -, so

$$n = \underline{9 - 4} + 6 - 1$$
$$n = \underline{5 + 6} - 1$$
$$n = 11 - 1$$
$$\boxed{n = 10}$$

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

(iii) a+=b*=c-=5 where a=3 b=5 c=8

Output = 1

Ans First c=5 is evaluated as $c = c - 5$
$= 8 - 5$
$= 3$

Second b*=c is evaluated as $b = b * c = 5 * 3$
$= 15$

Third a+=b is evaluated as $a = a + b$
$= 3 + 15$

(iv) $\underline{100/20} < = 10 - 5 + 100\%5 - 20 = 18$

$\Rightarrow 5 < = 10 - 5 + \underline{100\%5} - 20$

$\Rightarrow 5 < = \underline{10 - 5} + 0 - 20$

$\Rightarrow 5 < = \underline{5 + 0} - 20$

$\Rightarrow 5 < = \underline{5 - 20}$

$\Rightarrow 5 < = -15$

As $5 < = -15$ is false, the output will be zero.

Expression:- An expression is a formula in which operands are linked to each other by the use of operators to compute a value. An operand can be a function reference, a variable, an array element or a constant.

Ex:- a - b;

(-) is an operator
a, b are operand.

- Arithmetic expression
- Relational expression
- Logical expression
- conditional expression

Q6 What is the output of

```
int main()
{
int c = 2^3;
printf("%d", c);
}
```

Output:- 1

imp.

Que What is type conversion? Explain with example? (2016-17)

'or'

Differentiate implicit and explicit conversion with example. (2017-18)

Solⁿ When one data type converted into another data type by a compiler is called as type conversion. This is classified into two types

1. Implicit type conversion (Automatic)
2. Explicit type conversion (manual)

Difference:-

Implicit Type Conversion	Explicit Type conversion
1. It is automatic type conversion.	It is manual type conversion.
2. It is performed in lower to higher data type only	It can be performed in any order
3. float i ; i = 5.0/2 = 2.5 Here 5.0 belong to double data type 2 belongs to int data type so 2 get converted in to double before execution	float i ; i = (int) 5.0/2 = 2.0 5.0 belongs to double data type 2 belongs to int data type But 5.0 get converted in to int before execution.

imp.

Qu 8 What is difference between type casting and type conversion? (2019-20)

Ans

Type casting	Type conversion
(1) A data type is converted into another data type by a programmer using casting operator.	(1) A data type is converted into another data type by a compiler.
(2) Type casting can be applied to compatible data types as well as incompatible data types	(2) whereas type conversion can only be applied to compatible data types.
(3) Casting operator is needed in order to cast the a data type to another data type.	(3) There is no need for a casting operator.
(4) The destination data type may be smaller than the source data type, when converting the data type to another data type.	(4) The destination data type can't be smaller than source data type.
(5) It takes place during the program design by programmer	(5) It is done at the compile time.
(6) Type casting is more efficient and reliable	(6) Less efficient and less reliable
(7) Used in coding and competitive programming works	(7) Less used in coding it might cause incorrect answer.

Que9 write short note on various decision / selection control instruction used in C? (2020-21)

'or'

Explain the two way selection (if, if-else, nested if-else, cascaded if-else in C language with syntax. (2015-16)

Ans The control instruction which help to check some instruction, condition and take decision about what to execute is known as decision or selection control instruction.

There are four types of branch control instruction.

1. if else
2. conditional operator
3. goto statement
4. switch statement

1) if else control instructions

different type of if statement
(a) simple if
(b) if else
(c) nested if else
(d) if-else ladder

(a) Simple if :- It is used for executing a block of code when the condition is true, here false case is not defined.

Syntax:-
```
if (condition)
{
    Block of code;
}
```

The statement if tells the compiler that, what follows is a decision control instruction.

The zero value is false & non-zero is true.

Jump

Example:- a program to print the greatest of 3 numbers.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int a, b, c;
  clrscr();
  printf("\n Enter 3 Numbers: \n");
  scanf("%d %d %d", &a, &b, &c);
  if (a>b && a>c)
  {
   printf("%d "is largest", a);
  }
    if (b>a && b>c)
  {
   printf("%d is largest", b);
  }
    if (c>a && c>b)
  {
   printf("%d is largest", c);
  }
    getch();
  }
```

(b) if else statement: It is used for executing a block of code when the condition is true, here false case is also defined.

Syntax:-
```
if (condition)
{
  True block of code;
}
else
{
  false block of code;
}
```

- Statement used to match the corresponding if and else and pair of braces
- The zero value is false & non-zero is true.

Example:- A c program to check year is leap year or not (2016-17)

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int yr;
  clrscr();
  printf("\n Enter year:\n");
  scanf("%d", &yr);
  if(yr%400==0||(yr%4==0 && yr%100!=0)
  {
    printf("\n leap year\n");
  }
  else
```

```
{
  printf("\n Not leap year\n");
}
   getch();
}
```

(c) Nested if else statement: It is used for executing a block of code when there are multiple conditions written seprately using if.

Syntax:-
```
if (condition1)
{
  if(condition2)
  {
    True block of code;
  }
  }
   else
  {
    false block of code;
  }
```


① Statement need to be match the corresponding if and else and pair of braces.

EX:- Admission to the course has following condition
(i) maths >=60 (ii) Physics >= 50 (iii) chem >= 40
(iv) Total in 3 subject >= 200 or Total in maths & physics >=150
WAP that accepts the marks of three subject & determine the candidate is eligible or not.

Sol^n
```
# include <stdio.h>
# include <conio.h>
```

```
void main()
{ int m, p, c, tmpc, tmp;
  clrscr();
  printf("\n enter marks of m, p, c \n");
  scanf("%d %d %d", &m, &p, &c);
    tmpc = m+p+c;
    tmp = m+p;
  if (m>=60 && p>=50 && c>=40)
  {
    if( tmpc>=200 || tmp>=150)
    {
      printf("\n Eligible \n");
  3
    else
     {
      printf("\n Not Eligible \n");
    3
  3
    else
     {
       printf("\n Not Eligible \n");
     3
  getch();
 3
```

(d) **if else ladder statement** :- It is used for executing a block of code when there are multiple conditions written seprately using if statement and only one condition among them shall be true.

Syntax:-
```
if (condition 1)
    True block of code 1;
else if (condition 2)
    True block of code 2;
else if (condition 3)
    True block of code 3;
else
    False block of code;
```

Example 1: WAP in C to print the grade of student. (2016-17, 2015-16)

Soln
```
#include <stdio.h>
#include <conio.h>
void main()
{
    int pc;
    clrscr();
    printf("\n Enter percentage \n");
    scanf("%d", &pc);
    if(pc<0 || pc>100)
    printf("\n Invalid percentage \n");
    else if (pc>=80)
    printf("\n A grade \n");
    else if (pc>=60)
    printf("\n B grade \n");
    else if (pc>=40)
    printf("\n C grade \n");
    else
        printf("\n F grade \n");
    getch();
}
```

imp.
Ex:2 WAP to find the largest of three numbers using nested if else.

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int a, b, c, lar;
  printf ("enter value of a, b, c");
  scanf ("%d %d %d", &a, &b, &c);
  if (a > b)
    {
      if (a > c)
        lar = a;
      else
        lar = c;
    }
  else
    {
      if (b > c)
        lar = b;
      else
        lar = c;
    }
  printf ("largest is %d", lar);
  getch();
}
```

Ex: 3 WAP to check whether the given character is upper case, lower case, numeric or symbol. (2014-15, 2017-18)

Soln
```
# include <stdio.h>
# include <conio.h>
  void main()
  {
    char n;
    clrscr();
    printf("\n enter character : \n");
    scanf("%c", &n);
    if (n>=65 && n<=90)
    printf("\n Upper case character \n");
    else if (n>=97 && n<=122)
    printf("\n Lower case character \n");
    else if( n>=48 && n<=57)
    printf("\n Numeric character \n");
    else
     printf("\n symbol \n");
     getch();
    }
```

Ex: 4 Draw a program to find the quardent for the given coordinate using if else ladder. (2019-20)

Soln
```
# include <stdio.h>
  int main()
  {
    int a,b;
    scanf("%d %d", &a, &b);
    if (a>0 & b>0)
      printf("1st quadrant");
```

```
else if (a<0 && b>0)
 printf ("IInd quadrant");
else if (a<0 && b<0)
 printf ("IIIrd quadrant");
else if (a>0 && b<0)
 printf ("IVth quadrant");
 else
    printf ("origin");
    return 0;
```

O/P: -2
-3
IIIrd quadrant

Q11) What is the difference between if else statement & conditional operator.

Ans

If else	conditional operator
① It is a statement & does not return a value	① It is an expression & return a value.
② In if else, else can be optional	② In ? : all 3 expression are necessary.
③ Prefer where multiple statements are within block	③ Prefer where single statement are within expression.
Syntax:- if (condition) { True block of code; } else { false block of code; }	Syntax:- exp1 ? exp2 : exp3 Exp1 is true then Exp2 is executed otherwise Exp3 is executed.

Q11 What is the advantage and disadvantage using the goto statement? (2016-17, 2017-18)

Adv of goto:- → (Advantage)

- C support goto statement to branch unconditionally from one point to another in the program.
- The goto statement requires a label in order to identify the place.
- A label is any valid name that must be followed by a colon.
- It is used to come out of nested looping blocks.

Example:-

```
goto label;                label: statement
label: statement           goto label;
Forward jump               Backward jump
```

Disadvantage:- The use of "goto statement is highly discouraged as it makes the program logic very complex.

- use of goto makes the task of analyzing and verifying the correctness of programs.
- use of goto can be simply avoided using break & continue statements.

V. imp

Q12 Write the concept of switch statement. Write a program to draw the calculator using switch statement in c language? (2016-17, 2017-18)

Ans The control statement that allows us to make a

decision from the number of choices is called the switch case statement.

Rules for switch:-

1. The switch case must be constant or a constant expression.
2. The case label must be valid & unique.
3. case label must end with colon(:) and each statement with semi colon(;)
4. case label can be int or char constant but it can't be float.
5. Using break and default is optional.

Syntax:-

```
switch (Choice Variable)
{
  case value 1:
    block 1;
    break;
  case value 2:
    block 2;
    break;
  default:
    block n;
}
```

Ex:- WAP to design calculator.

```
#include <stdio.h>
#include <conio.h>
  void main()
```

```
{
  int a, b, c, ch;
  clrscr();
  printf("\n Enter 2 numbers: \n");
  scanf("%d %d", &a, &b);
  printf("\n Enter 1 for addition: \n");
  printf("\n Enter 2 for subtraction: \n");
  printf("\n Enter 3 for multiply: \n");
  printf("\n Enter 4 for division: \n");
  printf("\n Now Enter Your choice: \n");
  scanf("%d", &ch);
  switch(ch)
  {
  case 1: c = a+b;
          printf("Sum is %d", c);
          break;
  case 2: c = a-b;
          printf("Sub is %d", c);
          break;
  case 3: c = a*b
          printf("Mul is %d", c);
          break;
  case 4: c=a/b;
          printf("Div is %d", c);
          break;
  default: printf("\n wrong input \n");
  }
    getch();
}
```

Q) WAP to find the value of y for a particular value of n. The a, n, b, n is input by user. (2014-15)

If n=1 $y = ax \% b$ (2018-19, 2020-21)

If n=2 $y = ax^2 + b^2$

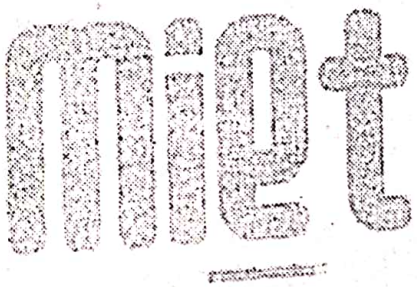
If n=3 $y = a - bx$

If n=4 $y = a + x/b$

Sol^n:-

```
# include <stdio.h>
# include <conio.h>
  void main()
  {
   int n, a, b, x, y;
   clrscr();
   printf("\n enter value of a, b, x \n");
   scanf("%d %d %d", &a, &b, &x);
   printf("\n The choices are as follows \n");
   printf("\n 1: y = a * x % b \n");
   printf("\n 2: y = a * x * x + b * b \n");
   printf("\n 3: y = a - b * x \n");
   printf("\n 4: y = a + x/b \n");
   printf("\n Now enter your choice \n");
   scanf("%d", &n)
   switch(n)
   {
    case 1: y = a * x % b;
    printf("value = %d.", y);
    break;
```

```
case 2: y = a * n * n + b * b ;
 printf ("value = %d ", y);
  break ;
 case 3: y = a - b * n ;
 printf ("value = %d ", y );
 break ;
case 4 : y = a + n / b ;
 printf ("value = %d ", y );
 break ;
default : printf ("\n wrong choice \n");
}
  getch();
}
```



5 Year's

University Paper Questions

(AKTU Question Bank)

5 Years AKTU University Examination Questions		Unit-2	
S. No	Questions	Session	Lecture No
1	Evaluate the following expression x = a-b/3 + c*2-1 When a = 9, b = 12, and c = 3	2014-15(odd)	10
	What is an expression? Evaluate the following expressions i) 100 % 20 < = 20 -- 5 + 100 % 10 -- 20 = = 5 > = 1 != 20 ii) a + = b * = C - = 5 where a=3 b=5 and c=8	2015-16(odd)	10
2	Evaluate the following expression 9-12/6*(2-1)	2016-17(even)	10
3	Define i)variable ii) Constant iii) Associativity iv)Precedence	2017-18(Odd), 2019-20(Odd), 2020-	10
4	What is the output of int main(){int c=2^3;printf("%d",c);	21(Odd), 2016-17(Even)	11
5	List all the operators used in C. Give examples	2018-19(Odd)	11
6	What are the different types of relation operator? Discuss each with example	2019-20(Odd), 2017-18(Odd), 2014-15(Even)	11
7	What is an assignment statement? Give the general form of an assignment statement.	2019-20(Odd), 2014-15(Even)	11
8	Explain the following operators in C language i) Relational ii) Logical iii) Conditional	2016-17(Even)	11
9	What is an operator? Explain the arithmetic, relational, logical, and assignment operators in C language.	2015-16(Odd), 2019-20(Odd), 2020-21(Odd)	11
10	What are bitwise operators? How they are different from logical operators	2017-18(Odd)	10
11	What is conditional operator? Explain with suitable example	2018-19(Odd)	10
12	Write a C program to find the largest of three numbers using ternary operator	2016-17(Odd)	10
13	Differentiate post increment and pre increment with example	2016-17(Odd)	10
14	Differentiate between precedence and associativity.	2016-17(Odd	12
15	What is type conversion? Explain with example.	2016-17(Even)	12
16	Differentiate implicit and explicit conversion with example	2017-18(Even)	12
17	What is difference between type casting and type conversion?	2019-20(Odd)	12
18	Write short note on various decision/selection control instruction used in c.	2020-21(Odd)	13
19	Explain the two way selection (if, if-else, nested if-else, cascaded if else in C language with syntax.	2015-16(Even)	13
20	What is the difference between if else statement & Conditional Operator.	2017-18(Even)	13
21	Write C program to check number is even or odd.	2015-16(Odd)	13
22	Design and develop a C program to read a year as an input and find whether it is leap year or not. Also consider end of the centuries.	2015-16(Odd), 2019-20(Odd), 2017-18(Even)	14
23	What is the advantage and disadvantage of using the goto statement & if-else-if ladder?	2017-18(Odd), 2016-17(Even)	14

B. Tech I Year [Subject Name: Programming for Prob. Sol.]

24	Write a program to find whether the year is leap year or not, without using logical operator.	2016-17(Even)	14
25	Write the program to print the division or grade of student for given percentage.	2016-17(Odd), 2015-16(Even)	15
26	Draw the program to find the quadrant for the given coordinates using if else ladder.	2019-20(Odd)	15
27	WAP to check whether the given character is upper case, lower case, numeric or symbol.	2014-15(Odd), 2017-18(Odd)	
28	Write a program to find the value of y for a particular value of n. The a, x, b, n is input by user If n=1 y=ax%b if n=2 y=ax2+b2 if n=3 y=a-bx if n=4 y=a+x/b	2014-15(Odd), 2018-19(Odd), 2020-2021(Odd	16
29	Write a program to draw the calculator using switch statement in c language.	2016-17(Odd), 2017-18(Odd)	16