CBCS SCHEME

USN | | | | | | | | | | |

BPOPS103/203

First/Second Semester B.E./B.Tech. Degree Examination, June/July 2024
Principles of Programming using C

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	Define Computer. Explain the various types of computer.	10	L2	CO1
	b.	Explain the basic structures of C program in detail. Write a sample program to demonstrate the components in the structure of C program.	10	L2	CO2
		OR			
Q.2	a.	Explain scanf( ) and printf( ) functions in C language with syntax and example.	08	L2	CO2
	b.	What is variable? Explain rules for constructing variable in C. Give example for valid and invalid variable.	06	L2	CO2
	c.	Illustrate the flowchart and write a C program which takes as input p, t, v compute the simple interest and display result.	06	L2	CO2
		Module – 2			
Q.3	a.	Explain the following operators in 'C': i) Relational ii) Logical iii) Conditional iv) Bitwise.	08	L2	CO2
	b.	Explain for loop statement with syntax and example program.	06	L2	CO2
	c.	Write a C program to simulate simple calculator that performs arithmetic operations using switch statement. Error message should be displayed if any attempt is made to divide by zero.	06	L2	CO3
		OR			
Q.4	a.	Explain if, if-else, nested if and cascaded if-else statements with syntax and example.	08	L2	CO2
	b.	Write a C program that takes three coefficient (a, b, c) to calculate roots of quadratic equation, print all possible roots with appropriate messages for a set of coefficients.	06	L2	CO5
	c.	Explain break and continue statements with respect while, do-while and for loops.	06	L2	CO2
		Module – 3			
Q.5	a.	Define function. Explain categories of user defined functions.	10	L2	CO4
	b.	Define two-dimension array. Write a C program to multiply 2 matrix by ensuring their multiplication compatibility.	10	L2	CO3
		OR			
Q.6	a.	Explain function call, function definition and function prototype with syntax and example for each.	10	L2	CO4
	b.	Write a C program to implement Binary search for integers.	05	L2	CO3
	c.	What is Recursion? Write a C program to compute factorial of number using recursion.	05	L2	CO3
		Module – 4			
Q.7	a.	Define string. Explain any four string manipulating functions with example.	10	L2	CO3
	b.	Write a C program to concatenate two strings without using built-in function strcut( ).	05	L2	CO3
	c.	Explain string unformatted input/output functions with example.	05	L2	CO3

		OR			
Q.8	**a.**	Define pointer. Explain pointer variable declaration and initialization with suitable example.	**08**	**L2**	**CO3**
	b.	Explain pass by value and pass by address with example.	**04**	**L2**	**CO3**
	c.	Write a C program using pointers to compute sum, mean, standard deviation of all elements stored in an array of n real numbers.	**08**	**L2**	**CO3**
		Module – 5			
Q.9	**a.**	Explain structure declaration and how structure member are accessed with example.	**10**	**L2**	**CO3**
	b.	Implement a structure to read, write and compute average marks and the students scoring above and below average of class N students.	**10**	**L3**	**CO5**
		OR			
Q.10	**a.**	Compare between structure and union with syntax and example.	**06**	**L2**	**CO3**
	b.	Explain fopen( ), fclose( ), fscanf( ) and fprintf( ) with syntax and example program considering all above functions.	**10**	**L2**	**CO4**
	c.	What are enumeration variable? How are they declared?	**04**	**L2**	**CO3**

* * * * *

2303BP0PS10357497

Visvesvaraya Technological University

Belagavi, Karnataka - 590 018.

Scheme & Solutions Signature of Scrutinizer

Subject Title : Principles of Programming Using C **Subject Code :** BPOPS103

Question Number	Solution	Marks Allocated
1a.	Module - 1 Computer - Device that computes, especially a programmable electronic machine that performs high-speed mathematical / logical operations that assembles, stores, correlates. → 2M Types: ① Mainframes computers ② Mini computers ③ Microcomputers / Personal computers } Explain 8M	10M

1b.	Structure of C program:
	Preprocessor directive ← 3M
	Global variable ← Example 3M
	main()
	{
	" body of main
	}
	← Explation 4M
	user-defined functi
		10M

Subject Title : Principles of Programming Using C Subject Code : BPOPS103

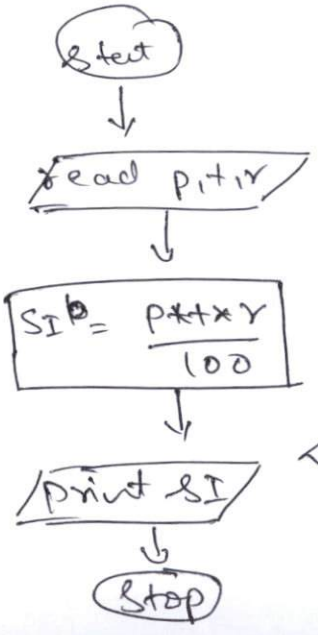
Question Number	Solution	Marks Allocated

2a.	scanf():
	Syntax:
	n= scanf ("Formed string", address list);
	Explation : ← 2M
	printf():
	n = printf("Formatstring", list of variable);
	Explation ← 2M
	main()
	{ int a, float b, char c;
	printf("Enter a, b & c");
	scanf ("%d %f %c", &a, &b, &c);
	printf("A=%d\n B=%f\n C=%c", a, b, c);
	}
	← 4M
		8M

2b.	variable :- its a name given to address of the memory. ← 1M Rules: ① It should begin with letter / underscore (_) ② No extra symbol expect (_) should not be used. ③ Keywords cannot be used as variables ④ letter / underscore can be followed by digits. ←	3M

Subject Title : Principles of Programming Using C Subject Code : BPOPS103

Question Number	Solution	Marks Allocated
	Valid variables: sum, sum_add, aveg. Not valid variables: sum+a, 09ab, #class. ← 2M	6M

2c.	program:
	main()
	{
	float p,t,r,si;
	printf("Enter p,t&r");
	scanf("%f%f%f",&p,&t,&r);
	si=(p*t*r)/100;
	printf("SI=%f",si);
	}
	 
	 ← 6M
		6M

	module - 2	
3a.	Operator in C: i) Relational ii) Logical iii) Conditional iv) Bitwise List operator used Example for each ← 2*4 = 8M	8M

Subject Title : Principles of Programming Using C Subject Code : BPOPS103

Question Number	Solution	Marks Allocated

3b.	for loop :-
	syntax:
	for (exp1 ; exp2 ; exp3)
	{
	" Body of the loop
	}
	Explation ← 1M
	Example:
	for (i=0; i<5; i++)
	{ printf("%d \t", i); ← 2M
	}
	← 3M
		6M

3c.	C program: { Logical statements need to be added with complex program)
	Simple calculator:
	switch (ch)
	{ case '+': res = a+b
	break;
	case '-': res = a-b;
	break;
	case '*': res = a*b;
	break;
	case '/': if (b != 0)
	{ res = a/b
	} break;
	else
	printf(" Error: divid by zero");
	break;
	case '%': res = a%b
	break;
	default: printf("Illegal operato");
	exit(0);
	} ← 6M	

Subject Title : Principles of Programming Using C Subject Code : BPOPS103

Question Number	Solution	Marks Allocated
4a.	if statement if - else nested - if cascade if-else } Syntax + example each ← 2×4 = 8M	

4b.	program: [logical statement to be added & complet prgeam]
	Pgm
	d = b*b - 4*a*c
	if (d==0)
	{ printf("Roots are real & equal");
	root1 = -b/a;
	root2 = root1;
	printf("R1 = %f \n R2 = %f", root1, root2);
	}
	else if (d>0)
	{ printf("Root are Real & distic");
	root1 = (-b + sqrt(d)) / (2*a);
	root2 = (-b - sqrt(d)) / (2*a);
	ps("R1 = %f \n R2 = %f", root1, root2);
	}
	else
	{ printf("Roots are imaginery");
	rp = -b/(2*a);
	ip = (sqrt(fabs(d)) / (2*a);
	printf("R1 = %f + i%f", rp, ip);
	printf("R2 = %f - i%f", rp, ip);
	} ← 6M	

Subject Title : Principles of Programming Using C Subject Code : BPOPS103

Question Number	Solution	Marks Allocated
4 C.	**break:** encontered inside a while, do-while or for statements the statement is immediately terminated and comes out of the loop and encout followed statements.	

```
while (exp)
{
    statement-1;
    break;
    statement-2;
}
```

```
do
{
    statement-1;
    break;
    statement-2;
} while(exp)
```

```
for(exp1; exp2; exp3)
{
    statement-1;
    break;
    statement 2;
}
```

Ex:

```
int i=1;
for (i=1; i<5 ; i++)
{
    if (i==3) break;
    printf("%d\t ", i);
}
```

Output:

1 2

← 3M

Continue: these statement terminates the current iteration of while, do-while / for and resume execution back at the beginning of loop of body. with next iteration.

Subject Title : Principles of Programming Using C Subject Code : BPOPS103

Question Number	Solution	Marks Allocated

```
while (exp)
{
  stat-1
  continue;
  stat-2;
}
```

```
do
{
  stat-1
  continue;
  stat-n
} while (exp)
```

```
for (ep1; ep2; ep3)
{
  stat-1;
  continue;
  stat 2;
}
```

Example

```
int i=1;
for(i=1; i<=5; i++)
{ if (i==3) continue;
  printf("%d ", i);
}
```

Output:

1 2 4 5 ← 3M

6M

Module -3

5a. function :- group of statement outside main to do some specific task ← 1M

Categories:

① With parameter & with return value
② With parameter & with no return value
③ With no parameter & with return value
④ With no parameter & with no return value

List ← 1M

explain with example

4X2 = 8M

10M

Subject Title : Principles of Programming Using C Subject Code : BPOPS103

Question Number	Solution	Marks Allocated

5b.	2D-array: an array with 2 sets of square brackets [][].
	* two Dimension array used when data items are arranged in row-wise & colum-wise ← 3M
	
	progam: Matrix multiplication.
	{ logical statement to read Matrix, to print matrix & to ensure compatibility to be added to complet progam}. ← 2M
	main Logic :
	for (i=0; i<m; i++)
	{
	for (j=0; j<q; j++)
	{
	c[i][j] = 0;
	for (k=0; k<m; k++)
	{
	c[i][j] = c[i][j] + a[i][k] * b[k][j];
	}
	}
	}
	← 3M
	Output :
	① for not multiplicat. {Negative case}
	② for proper output. ← 2M
		10M

Subject Title: Principles of Programming Using C Subject Code: BPOPS103

Question Number	Solution	Marks Allocated
6a.	Function call Function defination Function prototype } syntax + example each → 3 + 3½ + 3½	10M

6b.	program: {Logical statements need to be added with complet program}
	Binary search:
	low = 0, high = n-1;
	while (low <= high)
	{ mid = (low + high)/2;
	if (key == a[mid])
	{ found = 1;
	break;
	}
	else if (key > a[mid])
	low = mid+1;
	else
	high = mid-1;
	} ← 5M	

6c.	Recursion:- Recusion function is a function that calls same or different itself during execution. ← 2M	

Subject Title: Principles of Programming Using C Subject Code: BPOPS103

Question Number	Solution	Marks Allocated

	program: factorial of n
	int fact (int n)
	{
	if (n==0)
	return 1;
	else
	return (n * fact(n-1));
	main ()
	{ int n, fact res;
	printf("Enter n");
	scanf("%d", &n);
	res = fact(n);
	printf("Factorial = %d", res);
	}
		← 3M
		5M

	Module- 4	
7a.	String:- Collection of characters marked by '\0' at the end.	← 2M
	Functions: strcpy(), strcmp(), strlen(), strncpy(), strcat(), strncat(), strrev(), strlwr(), strupr(). {any 4} syntax + example	4X2=8M 10M

Subject Title: Principles of Programming Using C Subject Code: BPOPS103

Question Number	Solution	Marks Allocated
7b.	program: Concatenate two strings. void concatenate (char [], char [])	

```
{
    int i=0;
    while (s1[i]!='\0')
        i++;
    int j=0;
    while (s2[j]!='\0')
    {
        s1[i] = s2[j];
        i++;
        j++;
    }
    s1[i]='\0'
}
```

[Logical statement (main()) need to be added to complet program]. ← 5M

7c. String unformatted input / output functions:

- gets(), getche(), putch() ← 2½
- gets(), puts() ← 2½

(Syntax + example) — 5M.

8a. pointer :- variable that stores the address of another variable. ← 2M.

Subject Title: Principles of Programming Using C Subject Code: BPOPS103

Question Number	Solution	Marks Allocated

Declaring of pointer variable:

General form:

data-type *pointer-name;

where

* → tells the variable is pointer variable.

pointer-name: as identifie/variable

data-type: int float, char.

Example:

int *p;

where p is a pointer to an integer variable. ← 3M

Initilization of pointer:

its very important to initilize a pointer

① Declear data variable

② Declear pointer variable

③ Assign address of data variable to pointer variable using & operato.

Ex:

```
int a;
int *ptr;
ptr = &a;
```

← 3M

8M

Subject Title: Principles of Programming Using C Subject Code: BPOPS103

Question Number	Solution	Marks Allocated
8b.	Pass by value: means when formal parameter changes actual parameter will not change.	

Example:

```
void exchange (int m, int n)
{
    int temp;
        temp = m;
        m = n.
        n = temp;
}
int main()
{
    int a,b;
        a=10, b=20;
    exchange(a,b)
    printf("A=%d \t B=%d",a,b);
}
```

Output;
A=10
B=20.

← 2M

pass by address: means when formal parameter changes actual parameter also changes.

```
void exchange(int *m, int *n)
{   int temp;
        temp = *m;
        *m = *n;
        *n = temp;
}
int main()
{   int a=10,b=20;
    exchange(&a,&b);
    printf("A=%d & B=%d", a,b);
}
```

← 2M

4M

Output : A=20, B=10.

Subject Title: Principles of Programming Using C Subject Code: BPOPS103

Question Number	Solution	Marks Allocated

8c.	program: [Logical statements along with complet program].
	
	ptr = a;
	for (i = 0; i < n; i++)
	{
	sum = sum + *ptr;
	ptr++;
	}
	mean = sum/n;
	
	ptr = a;
	for(i = 0; i < n; i++)
	{
	stdsum = stdsum + pow((*ptr - mean),2);
	ptr++;
	}
	std = sqrt(stdsum/n); ← 8M	

	Module-5	
9a.	structure declaration: → Syntax: + example program. ← 3M → Assessing member of structure [creating variable for structure] ← 3M → Initilizing the value for member of structure using (.) Dot operator with example ← 4M	10M.

Subject Title: Principles of Programming Using C Subject Code: BPOPS103

Question Number	Solution	Marks Allocated
9b.	Program: Structure to read, write & compute average.	

```
struct student
{
    char name[10];
    int rollno;
    int marks;
};
main()
{
    struct student s[10];
    printf("Enter no. of students\n");
    scanf("%d", &n);
    printf("Enter student details");
    for(i=0; i<n; i++)
    {
    printf("Enter Name:");
    scanf("%s", s[i].name);
    printf("Ent rollno:");
    scanf("%d", &s[i].rollno);
    printf("Ent marks:");
    scanf("%d", &s[i].marks);
}
// to find average
for(i=0; i<n; i++)
{
sum = sum + s[i].marks;
}
avg = sum/n;
for(i=0; i<n; i++)
{
    if(s[i].marks >= avg)
        countav++;
    else
        countbav++;
}
```

[Logical statement to be added to complete programs].

← 10M

Subject Title: Principles of programming Using C Subject Code: BPOPS103

Question Number	Solution	Marks Allocated
10 a.		

	Structure	Union
①	Struct to define a structure.	union is used to define a union
②	user can access individual members at a given time.	user can access only one member at a given time

③	syntax:
	struct strname
	{
	type v1;
	type v2;
	}
		union un_name
		{
		type v1;
		type v2;
		};
		← 6M

④	In case of a structure, there is a specific memory location for every input data member. Thus it can store multiple value of various members.	In this case of union, there is an allocation of only one shared memory for all inputs data members. Thus it can store one value at a time for all member.

10 b.	fopen() fclose() fprintf() fscanf()	syntax + program considering all 4 functions.	← 4X2=8M ← 2M 10M

Subject Title: Principles of Programming Using C Subject Code: BPOPS103

Question Number	Solution	Marks Allocated
10 c.	**Enumeration Variable:-** Enum in C is a special kind of datatype defined by the user. It consist of constant integers or integer that are given names by ausel. ← 2M Define enum:- enum enum-name { int con1, int con2, .. int conN }; Example: enum cars { BMW, Feeran, Jeep }; Here, dfaul value for constants are: BMW = 0, Feeran = 1, Jeep = 2. However, to chage default valul, you can define the enumes following: enum cars { BMW = 1, Feeran = 5, Jeep = 0 }; ← 2M 4M.	