

USN 

--	--	--	--	--	--	--	--	--	--



## Internal Assessment Test 3 – January 2023

Sub:	Principles of Programming using C				Sub Code:	22POP13	Branch:	CSE			
Date:	23/1/2023	Duration:	90 mins	Max Marks:	50	Sem / Sec:	I [All sections]			OBE	
<u>Answer any FIVE FULL Questions</u>									MAR KS	CO	RB T
1	Explain the various rules for forming identifier's names. Identify the following identifiers as valid or invalid identifiers? If name is invalid, explain why? (i) cmrit (ii) team_4 (iii) 5th_cross (iv) min-max (v) marks%					10	CO2	L2			
2	What is the purpose of a printf() and scanf() statements? Explain the formatted printf() and scanf() along with the respective examples.					10	CO2	L2			
3	Write a C program to read maths, english, kannada subject marks and find total, average and Percentage and print.					10	CO2	L3			
4	List the basic data types available in C. Explain with example. Mention size in bytes.					5	CO2	L2			
	Explain different types of files used in C program.					5	CO1	L1			
5	Explain the general structure of a C program with an example.					5	CO1	L1			
	Explain components of computer with block diagram.					5	CO1	L2			
6	Write a C program to swap two numbers without using temporary variable.					5	CO2	L3			
	Write a C program to accept length and width from user and find the area of rectangle and display.					5	CO2	L3			

CI

CCI

HOD

USN 

--	--	--	--	--	--	--	--	--	--



## Internal Assessment Test 3 – January 2023

Sub:	Principles of Programming using C				Sub Code:	22POP13	Branch:	CSE			
Date:	23/1/2023	Duration:	90 mins	Max Marks:	50	Sem / Sec:	I [All sections]			OBE	
<u>Answer any FIVE FULL Questions</u>									MAR KS	CO	RB T
1	Explain the various rules for forming identifier's names. Identify the following identifiers as valid or invalid identifiers? If name is invalid, explain why? (i) cmrit (ii) team_4 (iii) 5th_cross (iv) min-max (v) marks%					10	CO2	L2			
2	What is the purpose of a printf() and scanf() statements? Explain the formatted printf() and scanf() along with the respective examples.					10	CO2	L2			
3	Write a C program to read maths, english, kannada subject marks and find total, average and Percentage and print.					10	CO2	L3			
4	List the basic data types available in C. Explain with example. Mention size in bytes.					5	CO2	L2			
	Explain different types of files used in C program.					5	CO1	L1			
5	Explain the general structure of a C program with an example.					5	CO1	L1			
	Explain components of computer with block diagram.					5	CO1	L2			
6	Write a C program to swap two numbers without using temporary variable.					5	CO2	L3			
	Write a C program to accept length and width from user and find the area of rectangle and display.					5	CO2	L3			

CI

CCI

HOD

Course Outcomes		Module s cover ed	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
			O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CO1	Elucidate the basic architecture and functionalities of a computer with recognition of the hardware devices/parts.	1	3	2	2	2	2	-	-	-	2	-	-	2	2	-	2	-
CO2	Apply programming constructs of C language for the solution of the real-world problem	1,2	3	3	3	3	2	-	-	-	2	2	-	2	2	-	2	-
CO3	Explore user-defined data structures like arrays with implementation of searching and sorting algorithms in C Language	2,3	3	3	3	3	2	-	-	-	2	2	-	2	2	-	2	-
CO4	Develop Solutions to problems using modular programming constructs by user defined functions	2,3,4	3	3	3	3	2	-	-	-	2	2	-	2	2	-	2	-
CO5	Implement user-defined data structures like strut, union and pointers in C Language	2,3,4,5	3	3	3	3	2	-	-	-	2	2	-	2	2	-	2	-

COGNITIVE LEVEL	REVISED BLOOMS TAXONOMY KEYWORDS
L1	List, define, tell, describe, identify, show, label, collect, examine, tabulate, quote, name, who, when, where, etc.
L2	summarize, describe, interpret, contrast, predict, associate, distinguish, estimate, differentiate, discuss, extend
L3	Apply, demonstrate, calculate, complete, illustrate, show, solve, examine, modify, relate, change, classify, experiment, discover.
L4	Analyze, separate, order, explain, connect, classify, arrange, divide, compare, select, explain, infer.
L5	Assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize.

PROGRAM OUTCOMES (PO), PROGRAM SPECIFIC OUTCOMES (PSO)				CORRELATION LEVELS	
PO1	Engineering knowledge	PO7	Environment and sustainability	0	No Correlation
PO2	Problem analysis	PO8	Ethics	1	Slight/Low
PO3	Design/development of solutions	PO9	Individual and teamwork	2	Moderate/ Medium
PO4	Conduct investigations of complex problems	PO10	Communication	3	Substantial/ High
PO5	Modern tool usage	PO11	Project management and finance		
PO6	The Engineer and society	PO12	Life-long learning		
PSO1	Design and develop applications using different stacks of web and programming technologies.				
PSO2	Design and develop secure, parallel, distributed, networked, and digital systems.				
PSO3	Apply software engineering methods to design, develop, test and manage software systems.				
PSO4	Design and develop intelligent applications for business and industry.				

USN



## Internal Assessment Test 1 – January 2023

Sub:	Principles of Programming using C				Sub Code:	22POP13	Branch:		
Date:	23/1/2023	Duration:	90 min's	Max Marks:	50	Sem/Sec:	I/ A, B, C, D, E, F and G		
<b><u>Answer any FIVE FULL Questions</u></b>								OBE	
	<b><u>Answer any FIVE FULL Questions</u></b>						MARKS	CO	RBT
1	<p>Explain the various rules for forming identifiers names. identify the following identifiers as valid or invalid identifiers? if name is valid, explain why? (i)cmrit (ii)team_4 (iii) 5th_cross (iv)min-max (v)marks%</p> <p><b><u>Solution:</u></b></p> <p style="text-align: center;">C IDENTIFIERS</p> <p>Identifiers are used as the general terminology for the names of variables, functions and arrays. These are user defined names consisting of arbitrarily long sequence of letters and digits with either a letter or the underscore(_) as a first character.</p> <p>There are certain rules that should be followed while naming c identifiers:</p> <ul style="list-style-type: none"> <li>• They must begin with a letter or underscore (_).</li> <li>• They must consist of only letters, digits, or underscore.</li> <li>• No other special character is allowed. It should not be a keyword.</li> <li>• It must not contain white space.</li> <li>• It should be up to 31 characters long as only first 31 characters are significant.</li> </ul> <p>(i) cmrit is valid because it starts with alphabet</p> <p>(ii) team_4 is valid as it starts with alphabet and underscore is allowed in variable names.</p> <p>(iii) 5<sup>th</sup>_cross is invalid as variable name can not start with a number</p> <p>(iv) min-max is invalid as – symbol is not allowed in variable name</p> <p>(v) marks% is invalid as % symbol is not allowed in variable name declaration</p>						10	CO2	L2

2	<p>What is the purpose of a printf() and scanf() statements? Explain the formatted printf() and scanf() along with the respective examples.</p> <p><b>Solution:</b></p> <p>scanf()  //built-in/standard / formatted input function to read input  syntax: scanf(“format specifier”, address of input variable);  //format specifier: int-%d, float-%f, char-%c  // &amp; - address operator  Example: scanf(“%d”, &amp;a);</p> <p>printf()  To display a message</p> <p>Syntax: printf(“message”); //displaying any message</p> <p>To display a value</p> <p>Syntax: printf(“format specifier”, name of variable holding the value);  //format specifier: int-%d, float-%f, char-%c  Example: printf(“%d”, a);</p> <p>C language supports two formatting functions : printf () and scanf ().  printf() is used to convert text data stored in the program onto text stream for o/p to the monitor.  scanf() is used to convert text stream coming from keyboard to data values and stores them in program variables.  These are called as formatted Input and output functions because they have special format to read and write The text from streams and then converting them into binary stream.  Syntax of printf() : printf ( “ control string” , var1, var2 .....var n).  printf(“Hello welcome to C program”)  printf( “ %d %d %f ” , a , b , c), where a and b are integer variables and c is float variable.  %d and %f are known as format specifiers</p> <p>Syntax of scanf() :  scanf ( “ Control string” , &amp; var1, &amp; var2, .....,&amp; var n).  Format specifiers : For integer variables ----- %d  For float variables ----- %f  For character variables ----- %c</p> <p>These are called as formatted Input and output functions because they have special format to read and write The text from streams and then converting them into binary stream.  Syntax of printf() :  printf( “ control string” , var1, var2 .....var n).  printf(“Hello welcome to C program”)  printf( “ %d %d %f ” , a , b , c), where a and b are integer variables and c is float variable.</p>	10	CO2	L2
3	<p>Write a C program to read maths, english, kannada subjects marks and find total, average and percentage and print.</p> <p><b>Solution:</b></p> <pre>// Program to calculate total, average and percentage #include&lt;stdio.h&gt; int main() {</pre>	10	CO2	L3

```

int maths, English, kannada, Total;
float Average, Percentage;
printf("enter marks in subjects");
scanf("%d%d%d",&maths,&English,&kannada);
Total=maths+English+kannada;
Average=Total/3.0;
Percentage=(Total*100)/300.0;
printf("total=%d\n",Total);
printf("average=%f\n",Average);
printf("percentage=%f\n",Percentage);
return 0;
}
Output

```

```

enter marks in subjects
80
80
80
total=240
average=80.000000
percentage=80.000000

```

4 List the basic data data types available in C. Explain with example. Mention size in bytes. Explain different types of files used in C program.

10

CO1, CO2 L1,L2

**Solution:**

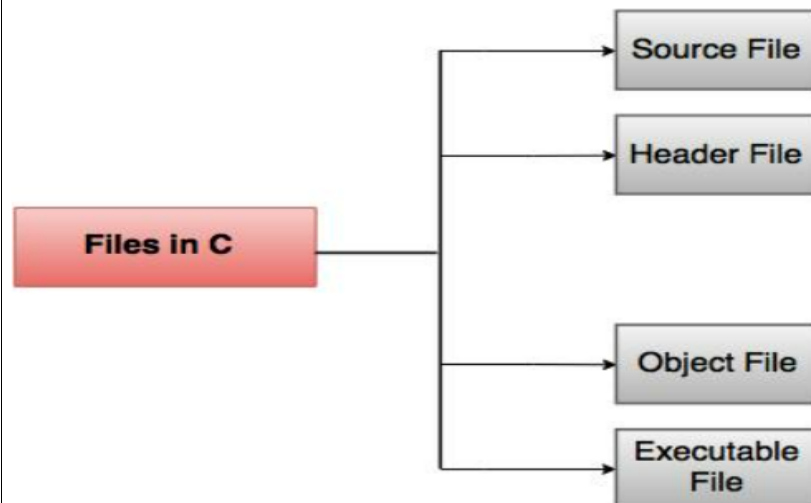
In C programming, data types are declarations for variables. This determines the type and size of data associated with variables.

C has the following basic built-in datatypes.

- int
- float
- double
- char

Data Type	Size
int	2 or 4 bytes
float	4 bytes
double	8 bytes
char	1 byte

**A C program uses four types of files as follows:**



**Source Code File**

This file includes the source code of the program.

The extension for these kind of files are '.c'. It defines the main and many more functions written in C.

main() is the starting point of the program. It may also contain other source code files.

**Header Files**

They have an extension '.h'. They contain the C function declarations and macro definitions that are shared between various source files.

**Object files**

They are the files that are generated by the compiler as the source code file is processed.

These files generally contain the binary code of the function definitions.

The object file is used by the linker for producing an executable file for combining the object files together. It has a '.o' extension.

**Executable file**

This file is generated by the linker.

Various object files are linked by the linker for producing a binary file which will be executed directly.

They have an '.exe' extension.

5	<p>Explain the general structure of C program with an example. Explain components of computer with block diagram</p> <p><b>Solutions:</b></p> <p>The components of the basic structure of a C program consists of 8 parts</p> <ol style="list-style-type: none"> <li>1. Document section</li> <li>2. Preprocessor/link Section</li> <li>3. Definition section</li> <li>4. Global declaration section</li> <li>5. Main function</li> <li>6. Local declaration section</li> <li>7. Logic part</li> <li>8. User-defined function section</li> </ol> <p>1. Documentation Section</p>	10	CO1	L1,L2
---	--	----	-----	-------

It is the section in which you can give comments to make the program more interactive. The compiler won't compile this and hence this portion would not be displayed on the output screen.

## 2. Preprocessor directives Section / link section

This section involves the use of header files that are to include necessarily program.

## 3. Definition section

This section is useful for constants declaration in C.

## 4. Global declaration Section

This section is used to define the global variables that are to be used in the programs. These variables can be used throughout the program.

## 5. Main function

It is the major section from where the execution of the program begins. The main section involves the declaration and executable section.

## 6. Local declaration section

This section is used to define the local variables that are to be used in the programs. These variables can be used in the function in which they are declared.

## 7. logic part

This section is used to write the logic of the program.

## 8. sub function section

When you want to define your function that fulfills a particular requirement, you can define them in this section.

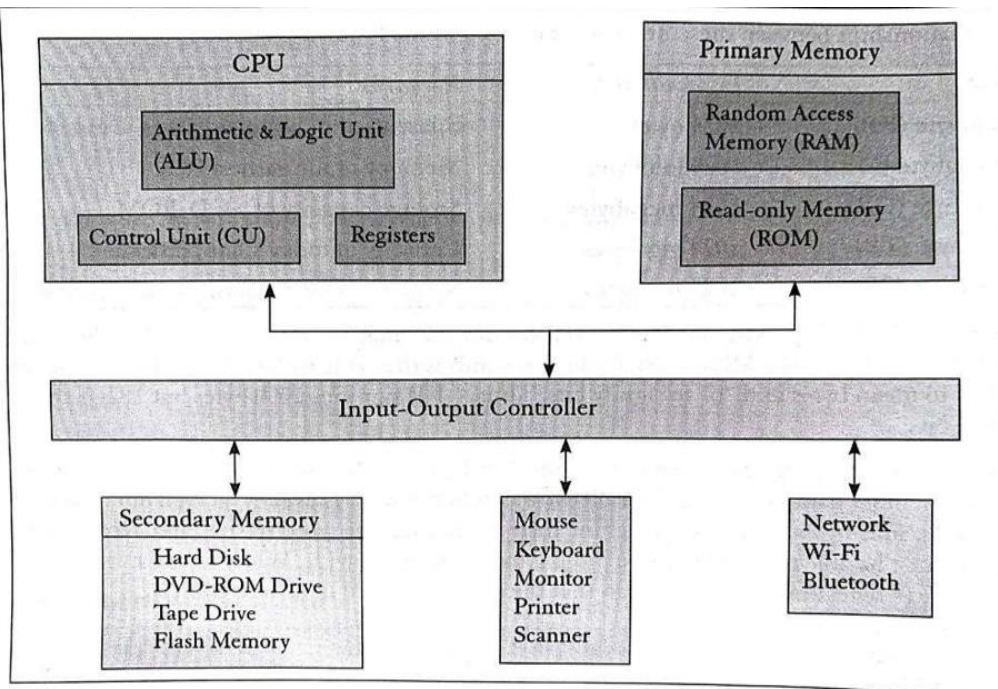
### **Example**

```
//area of a rectangle
#include<stdio.h>
int main()
{
    float length,breadth,area;

    printf("enter length of rectangle: ");
    scanf("%f",&length);

    printf("enter breadth of rectangle: ");
    scanf("%f",&breadth);

    area=(length*breadth);
    printf("AOR: %f\n",area);
    return 0;
}
```



### **THE CENTRAL PROCESSING UNIT (CPU)**

The CPU has evolved from a bulky vacuum tube based unit of the 1940s to a modern 5cm square chip that is commonly called the microprocessor, or simple processor. It comprises the following components

Arithmetic and Logic Unit (ALU)

Control Unit (CU)

Special purpose registers

A clock

### **PRIMARY MEMORY**

The primary memory which includes the following types:

Random Access Memory (RAM-SRAM and DRAM)

Read Only Memory (ROM, PROM, EPROM, EEPROM)

Cache Memory

CPU Registers

### **SECONDARY MEMORY**

The last couple of decades have seen the emergence of multiple types of storage devices.

Hard disk

Magnetic tape

CD-ROM (700 MB-less than 1 GB).

DVD-ROM (4.7 GB and 8.5 GB).

Blu-ray disk (27 GB and 50 GB).

Flash memory based on the EEPROM (1 GB to 128 GB).

The obsoleted floppy disk (1.2 MB and 1.44 MB).



	Input Device	Output Device		
<ul style="list-style-type: none"> <li>An input device is connected to accept data from users.</li> </ul>	<ul style="list-style-type: none"> <li>An output device is connected to provide information to users.</li> </ul>			
<ul style="list-style-type: none"> <li>Input devices are controlled by the user.</li> </ul>	<ul style="list-style-type: none"> <li>Output devices are controlled by the computer.</li> </ul>			
<ul style="list-style-type: none"> <li>Input devices translate user-friendly inputs to machine-understandable inputs.</li> </ul>	<ul style="list-style-type: none"> <li>Output devices translate machine-friendly outputs to user-understandable outputs.</li> </ul>			
<ul style="list-style-type: none"> <li>Examples: Mouse, Keyboard, Mic, Joystick, Scanner, Stylus, Trackball, etc.</li> </ul>	<ul style="list-style-type: none"> <li>Examples: Monitor, Printer, Speakers, Headphones, Projector, GPS, Plotter.</li> </ul>			
<p>6 Write a C program to swap two numbers without using temporary variable. Write a C program to accept length and width from user and find the area of rectangle and display.</p> <p><b>Solution:</b></p> <pre>// swapping without third variable #include&lt;stdio.h&gt; int main(){     int x,y;     printf("enter x and y values:");     scanf("%d%d",&amp;x,&amp;y);     x=x+y;     y=x-y;     x=x-y;     printf("After swap x=%d and y=%d",x,y);     return 0; }</pre> <p>Output</p> <pre>enter x and y values: 10 20 After swap x=20 and y=10</pre> <pre>//area of a rectangle #include&lt;stdio.h&gt; int main() {     float length,breadth,area;      printf("enter length of rectangle: ");     scanf("%f",&amp;length);      printf("enter breadth of rectangle: ");     scanf("%f",&amp;breadth);      area=(length*breadth);</pre>	10	CO2	L3	

```
printf("AOR: %f\n",area);  
return 0;
```

```
}
```

Output

```
enter length of rectangle:  
10  
enter breadth of rectangle: 20  
AOR: 200.000000
```