2.1 State whether the following statements are true or false.

a) Every line in a C Program Should end with a semicolon.**(false)**
b) The Closing Brace of the main() in a program is the logical end of the program.**(true)**
c) Comments cause the computer to print the text enclosed between /* and */ when executed.**(false)**
d) Every C program ends with END word.**(false)**
e) A printf Statement can generate only one line of O/P.**(false)**
f) The purpose of the header file such as stdio.h is to store the source code of program.**(false)**
g) A line in a program may have more than one statement .**(true)**
h) Syntax errors will be detected by the compiler.**(true)**
i) In C language lowercase letters are significant.**(true)**
j) Main() is where the program begins its execution.**(true)**
k) Every C program must have a least one user-defined function. **(false)**
l) Declaration section contains instruction to the computer.**(false)**
m) Only one function may be named main().**(true)**
n) Comments serve as internal documentation for programmers.**(true)**
o) In C , we can have comments inside comments.**(false)**
p) Use of comments reduce the speeds of execution of a program.**(false)**
q) A comment can be inserted in the middle of a statement.**(true)**

2.2 Fill in the blanks with appreciate words in each of the following statements.

a) Every program statement in C program must end with a **Semicolon(;)**.
b) The **printf** Function is used to display the output on the screen.
c) The **math.h** header file contains mathematical function.
d) The escape sequence character **\n** causes the cursor to move to the next lien on the screen .
e) C programs are written in lower case letters whereas upper case letters are mainly used to define **constant**.
f) C language offers several built-in **functions** that can be used in a program nu including the relevant header files.
g) **main** indicates the starting point C program .

2.3 Remove the semicolon(;) at the end of the printf statement in the following program and execute it .What is the O/P?

```
#include<stdio.h>

main()

{

/*…………..printing begins…………*/

Printf(“I see , I remember”);

/*………….printing ends…………..*/

}
```

ANSWER-: If you remove the semicolon at the end of the printf statement in the given C program and execute it, the program will result in a compilation error. The compiler expects a semicolon at the end of each statement. Without it, the code will not compile and run, so no output will be generated.

2.4 In the Following Program ,delete line-5 and execute the program . How helpful is the error message?

Sample program 2 -:

```
1   /*program edition*/
2   /*written By EBG*/
3   main()
4   {
5   int number;
6   float amount;
7
8    number = 100;
9
10  amount = 30.75 + 75.35;
11  printf(" %\n", number);
12  printf("%5.2f", amount);
13  }
```

ANSWER-: If you delete line 5 (int number;) and try to compile the program, you will get an error message related to the usage of number on line 11, where it is being used without being declared. The error message will typically indicate that number is undeclared or used without being defined.

error: ‘number’ undeclared (first use in this function)

2.5 Modify the sample Program 3 to display the following O/P.

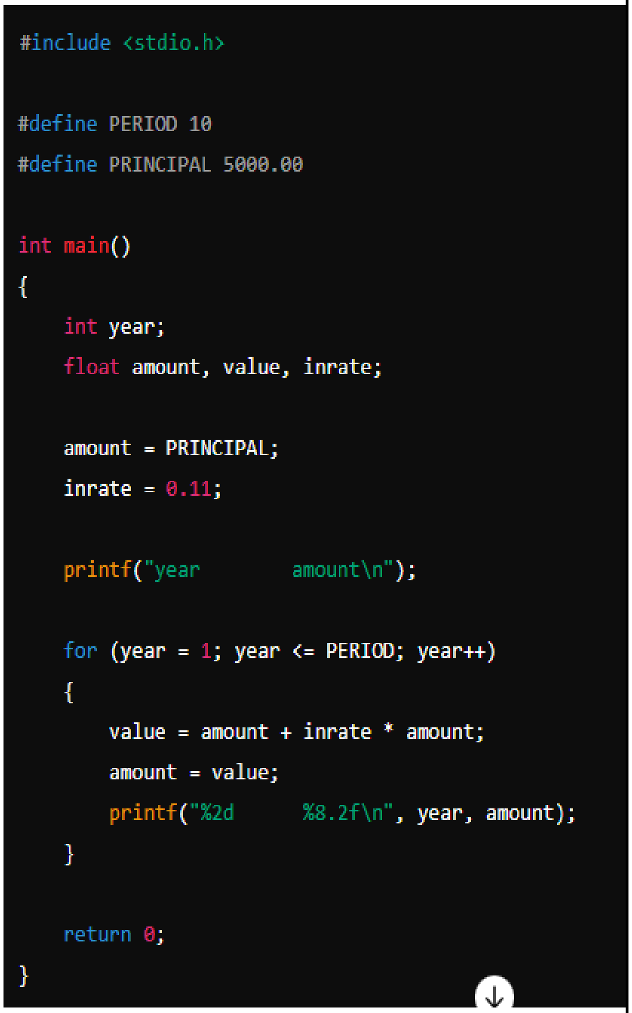
```
year        amount
1           5500.00
2           6160.00
```

```
_           ______

_           ______

10          14917.00
```

Modified Program as Question Requires



```c
#include <stdio.h>

#define PERIOD 10
#define PRINCIPAL 5000.00

int main()
{
    int year;
    float amount, value, inrate;

    amount = PRINCIPAL;
    inrate = 0.11;

    printf("year        amount\n");

    for (year = 1; year <= PERIOD; year++)
    {
        value = amount + inrate * amount;
        amount = value;
        printf("%2d      %8.2f\n", year, amount);
    }

    return 0;
}
```

Sample program 3

```
/*--------------------INVESTMENT PROBLEM-----------------*/

#define PERIOD 10

#define PRINCIPAL 5000.00

/*--------------------MIAN PROGRAM BEGINS-----------------*/

main()

{ /*----------DECLARATION STATEMENTS------------*/

Int year ;

Float amount , value,  inrate ;

/*--------------------ASSIGNEMNT STATEMENT -----------------*/

Amount = PRINCIPAL;

Inrate=0.11;

Year;

/*--------------------COMPUTATION STATEMENTS -----------------*/

/*------------------COMPUTATION USING while  LOOP ---------------*/

While(year <= PERIOD)

{ printf(“%2d     %8.2f\n”, year, amount);

  Value= amount + inrate * amount ;

Year= year +l;

Amount = value ;

                  }

/*----------------------while  LOOP ends----------------------*/

}

/*----------------------Programs ends--------------------*/
```

2.6 why and when do we use #define directive?

ANSWER:

The #define directive in C is used to create symbolic constants or macros, which are replaced by their values or expressions before compilation. This enhances code readability, maintainability, and allows for easier updates. It is commonly used for defining constants, avoiding magic numbers, creating function-like macros, and enabling conditional compilation.

Examples:

1. Defining Constants:

   ```
   #define PI 3.14159
   ```

2. Function-like Macros:

   ```
   #define SQUARE(x) ((x) * (x))
   ```

3. Conditional Compilation:

   ```
   #define DEBUG
   #ifdef DEBUG
   printf("Debug mode is ON\n");
   #endif
   ```

2.7 why and when do we use the #include directive ?

ANSWER:

When to use:

- When you need to use functions and definitions from standard libraries.
- When you want to include custom header files to organize and reuse code across multiple files.

Why use it:

- To access pre-defined functions and macros provided by libraries.
- To ensure code modularity and reusability by separating declarations and definitions.

```
#include <stdio.h> // Standard input/output library
int main() {
   printf("Hello, World!");
   return 0;
}
```

2.8 what does void main(void) mean?

ANSWER:

The declaration void main(void) in C or C++ is an incorrect and non-standard way to define the main function. The correct standard declarations for the main function are:

- int main(void) or int main(): A main function that takes no arguments and returns an integer.
- int main(int argc, char *argv[]): A main function that takes command-line arguments and returns an integer.

Explanation:

- void main(void) suggests that the main function does not return any value, which is non-standard. The return type of main should always be int as per the C and C++ standards, which allows the program to return a status code to the operating system.
- The parameter void indicates that the function takes no arguments, which is acceptable in terms of parameter specification but does not change the fact that main should return an integer.

2.9 Distinguish between the following pairs:

Main() and void main(void)

Int main() and void main()

ANSWER:

a) Main() and void main(void)

- Main(): Incorrect, non-standard capitalization; should be lowercase main.
- void main(void): Non-standard; main should return int.

b) int main() and void main()

- int main(): Standard and correct; returns an integer to indicate success or failure.
- void main(): Non-standard; main should return int to comply with C and C++ standards.

2.10 why do we need to use comment in programs?

ANSWER:

Comments in programs are for the following reasons:

1. **Code Readability**: They make the code easier to understand by explaining the purpose and functionality of code blocks.
2. **Documentation**: They provide a reference for future maintenance and for other developers who may work on the code.
3. **Debugging**: They help in temporarily disabling code without deleting it, aiding in debugging.

4. **Collaboration**: They improve communication among team members by providing context and explanations directly within the code.

2.11 why is the look of a program is important?

ANSWER:

=>The look of a program, which includes its formatting, structure, and readability, is important for several reasons:

1. **Readability**: Well-formatted code is easier to read and understand, reducing the chances of errors and misunderstandings.
2. **Maintainability**: Consistently structured code is easier to maintain, debug, and update, especially when revisited after some time or by other developers.
3. **Collaboration**: Clear and well-organized code facilitates collaboration among team members, making it easier for others to understand and contribute.
4. **Professionalism**: Neat and well-documented code reflects professionalism and attention to detail, which is crucial in a professional development environment.
5. **Error Reduction**: Proper indentation and spacing can help in identifying logical blocks of code and spotting errors or inconsistencies more easily.

2.12 where are blank spaces permitted in a C Program ?

ANSWER:

In a C program, blank spaces (whitespace) are permitted in the following places:

1. Between Keywords and Identifiers:

```
int main() { ... }
```

2. Between Operators and Operands:

```
int sum = a + b;
```

3. Between Function Names and Parameters:

```
printf("Hello, World!");
```

4. Between Statements:

```
int a = 5;
int b = 10;
```

5.Around Control Structures:

```
if (a > b) { ... }
```

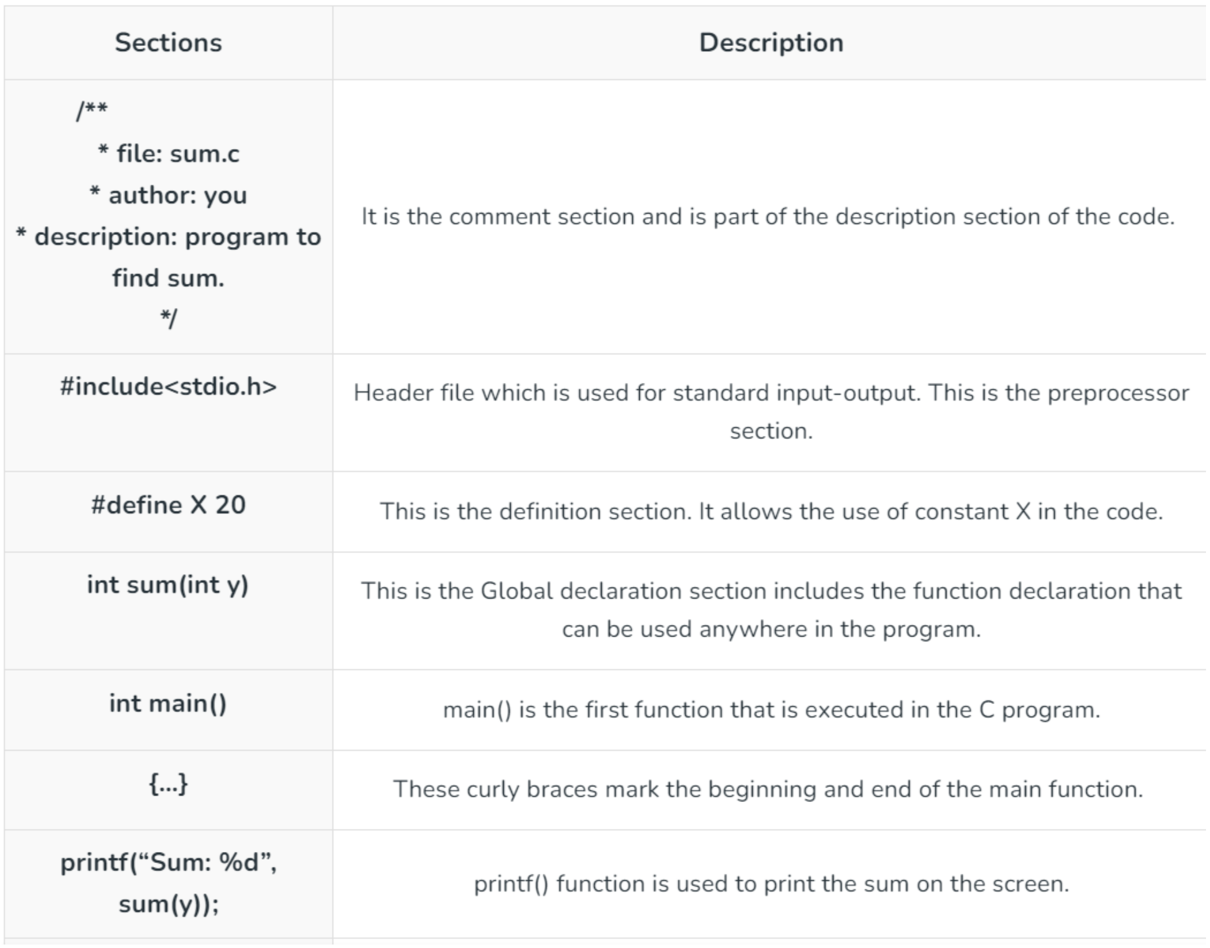
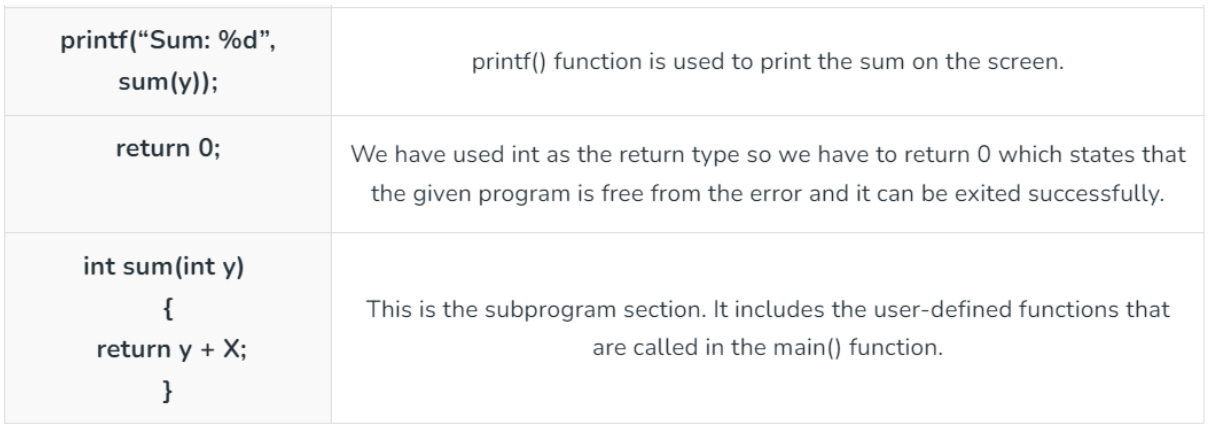
6. Within Parentheses:

```
for (int i = 0; i < 10; i++) { ... }
```

7. To Improve Readability:

```
int result = (a + b) * c;
```

2.13 Describe the Structure of C Program .

Sections	Description
/** * file: sum.c * author: you * description: program to find sum. */	It is the comment section and is part of the description section of the code.
#include<stdio.h>	Header file which is used for standard input-output. This is the preprocessor section.
#define X 20	This is the definition section. It allows the use of constant X in the code.
int sum(int y)	This is the Global declaration section includes the function declaration that can be used anywhere in the program.
int main()	main() is the first function that is executed in the C program.
{…}	These curly braces mark the beginning and end of the main function.
printf(“Sum: %d”, sum(y));	printf() function is used to print the sum on the screen.
printf(“Sum: %d”, sum(y));	printf() function is used to print the sum on the screen.
return 0;	We have used int as the return type so we have to return 0 which states that the given program is free from the error and it can be exited successfully.

int sum(int y)
{
return y + X;
}
	This is the subprogram section. It includes the user-defined functions that are called in the main() function.

2.14 Describe the process creating and executing a C Program under UNIX System .

Creating and Executing a C Program under UNIX System:

1. **Create the Source File: Write your C program using a text editor and save it with a \`.c\` extension.**

```
vi program.c
```

2. **Compile the Program: Use the \`gcc\` compiler to compile the source file into an executable.**

```
gcc program.c -o program
```

3. **Execute the Program: Run the compiled executable.**

```
./program
```

2.15 How do we implement MULTIPLE source program files?

Implementing Multiple Source Program Files in C:

1. **Create Source Files: Write separate \`.c\` files.**

- Example: \`main.c\`, \`utils.c\`

2. **Create Header Files: Define function prototypes in \`.h\` files.**

- Example: \`utils.h\`

3. **Include Header Files: Include the \`.h\` files in your \`.c\` files.**

```
// main.c
#include "utils.h"
```

4. **Compile Each File:**

```
gcc -c main.c
gcc -c utils.c
```

5. **Link Object Files:**

```
gcc main.o utils.o -o program
```

6. **Run the Program:**

./program