Contents

1 C++ Installation and Setup using Code::Blocks

1) **Download Code::Blocks**
   - You can download it using the following link or from the Teams Class. https://sourceforge.net/projects/codeblocks/files/Binaries/20.03/Windows/codeblocks-20.03mingw-setup.exe
   - Download the version with MinGW (includes a C++ compiler).
2) **Install Code::Blocks**
   - Run the installer and follow the on-screen instructions.
   - Choose Full Installation to include the compiler.
3) **Configure the Compiler**
   - Open Code::Blocks and go to Settings → Compiler.
   - Ensure that the selected compiler is GNU GCC Compiler.
4) **Create a New C++ Project**
   - Click File → New → Project.
   - Select Console Application → C++, then follow the setup wizard.
5) **Write and Run Your First Program**
   - Open main.cpp, write your C++ code, and press F9 to Build and Run.

Now, you're ready to code in C++ using Code::Blocks!

2 Basic Necessary Components of a C++ Program

A C++ program consists of several essential components that make it functional and structured. Below are the key components:

2.1 Preprocessor Directives

- These include header files that provide functionalities like input/output operations.
- Example:

```
#include <iostream>  //Includes the standard input-output stream library
```

2.2 Namespace Declaration

- The std namespace is used to access standard C++ library functions without prefixing std::
- Example:

```
using namespace std;
```

2.3 main () Function (Entry Point)

- Every C++ program must have a main() function where execution starts.
- Example:

```
int main()
{
        return 0;
}
```

2.4 Input/Output Statements

- Used for user interaction (cin for input, cout for output)
- Example

```
#include<iostream>
using namespace std;

int main()
{
    int value;
    cout<<"Enter a value "; //Console output
    cin>>value; //Console input
    cout<<"Entered value = "<<value;
    return 0;
}
```

2.5 Variables and Data Types

- Used to store and manipulate data
- See example in section 2.4 Input/Output Statements

2.6 Control Flow Statements

- Used to control execution flow (if, for, while, etc)
- Example

```
//if condition
#include<iostream>
using namespace std;
int main() {
    int value, condition=5;
    cout<<"Enter a value ";
    cin>>value;
    if (value > condition)
    {
        cout<<"The entered value is greater than 5";
    }
    else
    {
        cout<<"The entered value is less than 5";
    }
    return 0;
}
```

```
//for loop

#include<iostream>
using namespace std;

int main ()
{
    int range;

    cout<<"Enter range ";
    cin>>range;

    for (int i=1; i<=range; i++)
    {
        cout<<i<<endl;
    }
    return 0;
}
```

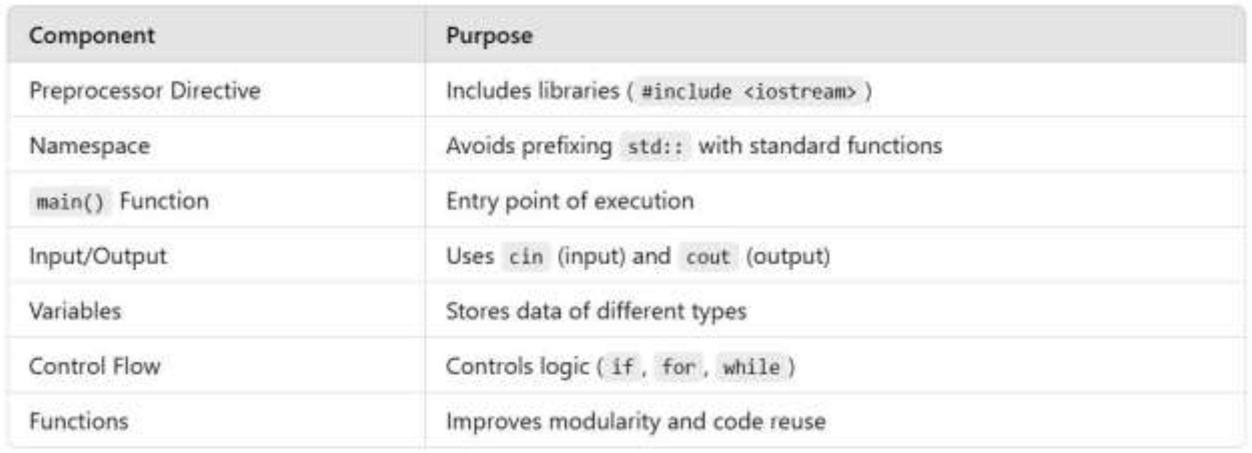
2.7 Functions

- Use to modularize code and improve readability.
- Example

```
//User define function

void greet()
{
    cout << "Welcome to C++!";
}
```

2.8 Summary of Components



Component	Purpose
Preprocessor Directive	Includes libraries (`#include <iostream>`)
Namespace	Avoids prefixing `std::` with standard functions
`main()` Function	Entry point of execution
Input/Output	Uses `cin` (input) and `cout` (output)
Variables	Stores data of different types
Control Flow	Controls logic (`if` , `for` , `while`)
Functions	Improves modularity and code reuse

3 TP

Write error-free C++ code for the given tasks and include it in a formal report. The report should also contain a screenshot of the program's output.

TP1) Write a C++ program that prints odd numbers up to a given range.

TP2) Write a C++ program that prints even numbers up to a given range.

TP3) Write a C++ program that prints an arithmetic series based on a given common difference and range.

TP4) Write a C++ program to check whether a given input is an odd or even number.

TP5) Write a C++ program to generate the Fibionacci Series up to the nth term.

Example:

Input: 10

Output: 0 1 1 2 3 5 8 13 21 34

TP6) Write a C++ program that calculates the factorial of a given positive integer. The program also checks whether the given input is a positive integer. Do not forget that 0! = 1.

TP7) Write a C++ program to check if a given number is prime or not.

Example:

Input: 25

Output: Not a prime.

TP8) Write a C++ program to print only prime numbers up to a given range.

TP9) Write a C++ program that inputs the base and power from the user and computes the result using a for loop.

TP10) Write a C++ program that prints the factorial of each number up to a given range.

Example:

Input: 5

Output:

Factorial of 1 is 1.

Factorial of 2 is 2.

Factorial of 3 is 6.

Factorial of 4 is 24.

Factorial of 5 is 120.

TP11) Write C++ code to find the GCD (Greatest Common Divisor) of two given numbers.

Example:

Input: 48, 18

Output: 6

TP12) For each task from TP1 to TP11, convert the code into user-defined functions. These functions should perform the core computation of the task but should neither take any input nor display any output. Use the `main ( )` function to handle all inputs and outputs. Additionally, ensure that the code includes input validation. Give an error message to the user for invalid inputs.

TP13) Write a C++ program that creates a custom function to rotate an array with the least number of swaps. Given an array of N integers and an integer K, rotate the array to the right by K positions using the minimum number of swaps. The solution should perform in-place swapping with minimum number of swaps without using any additional array.

TP14) Write a C++ program that defines a custom function to implement Bubble Sort algorithm for sorting the elements of a liner array but optimizes it to stop early when the array is already sorted.

TP15) Given the sorted linear array from TP14, remove all duplicate elements such that each element appears only once in the output, and return the new length of the array.

TP16) Write a C++ custom function that simulates an authentication process by prompting the user to enter a username and password. Use a do-while loop to ensure the input is repeated until valid credentials are provided.

TP17) Repeat TP16 by implementing recursive function to achieve the same results.