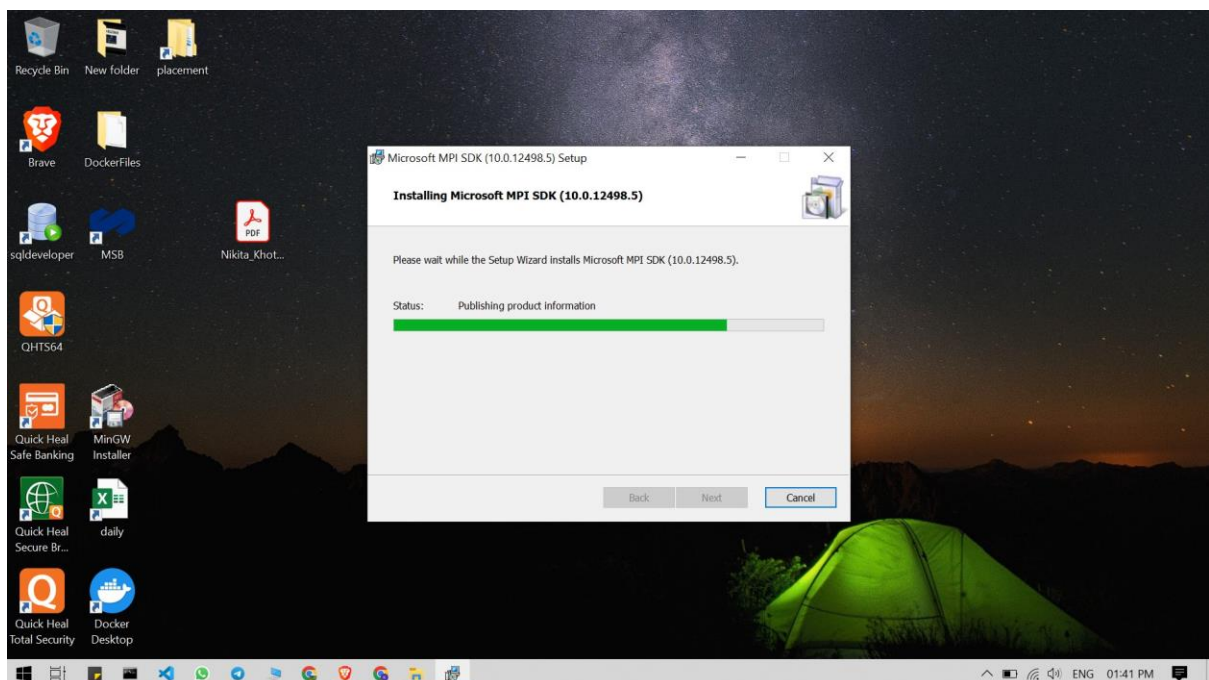**2020BTECS00041**
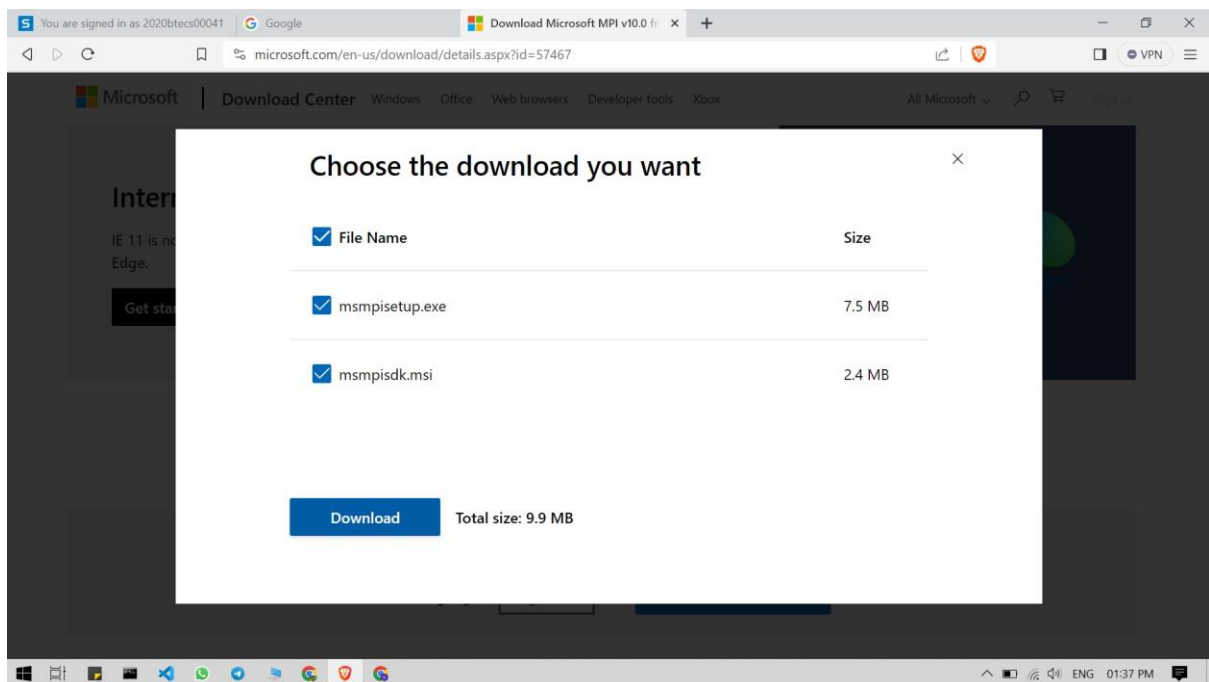**Nikita Shivchandra Khot**

High Performance Computing Lab
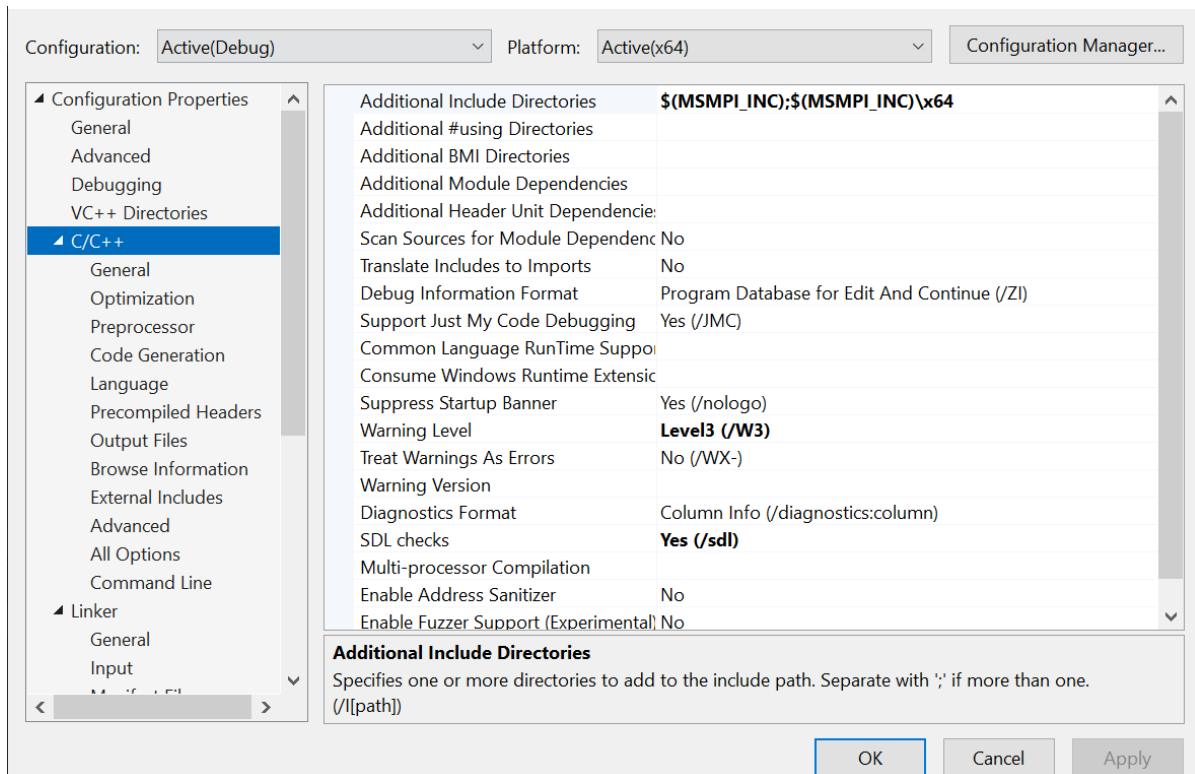## Practical No. 7
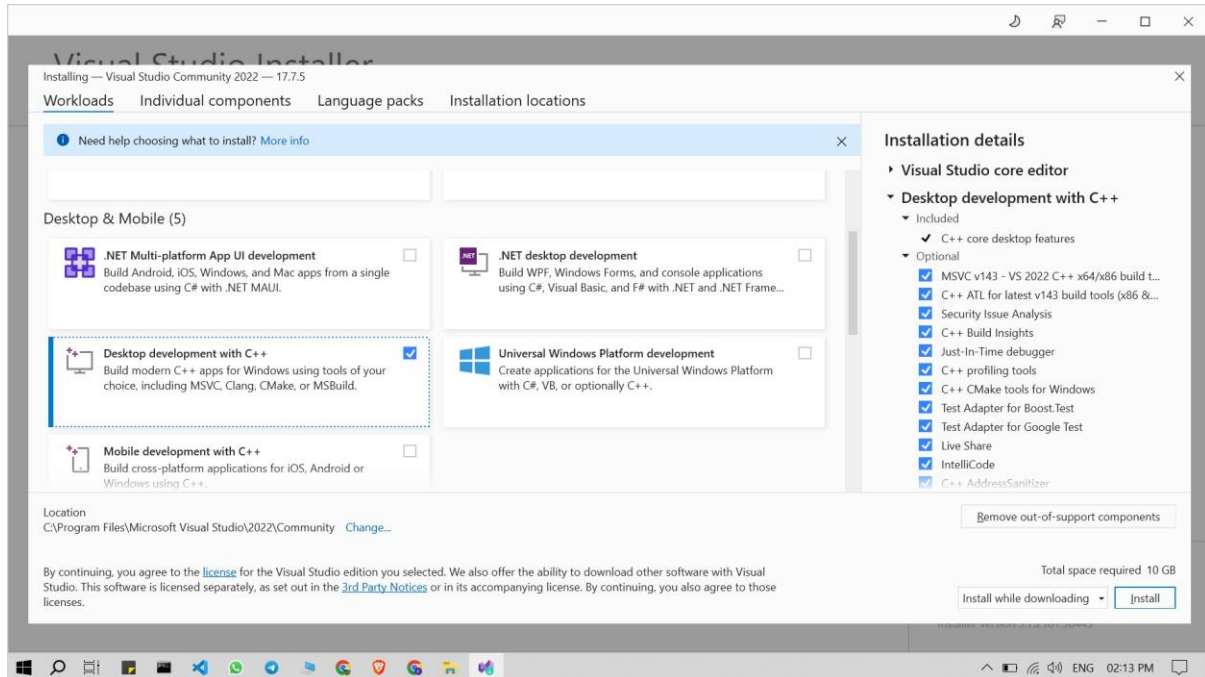
Title of practical: Installation of MPI & Implementation of basic functions of MPI
Installation of MPI

Visual Studio Installer

Installing — Visual Studio Community 2022 — 17.7.5

Workloads | Individual components | Language packs | Installation locations

ℹ Need help choosing what to install? More info

Desktop & Mobile (5)

- **.NET Multi-platform App UI development**
  Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.
- **.NET desktop development**
  Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...
- **Desktop development with C++** ✔
  Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.
- **Universal Windows Platform development**
  Create applications for the Universal Windows Platform with C#, VB, or optionally C++.
- **Mobile development with C++**
  Build cross-platform applications for iOS, Android or Windows using C++.

Installation details

▸ Visual Studio core editor
▾ Desktop development with C++
  ▾ Included
    ✔ C++ core desktop features
  ▾ Optional
    ☑ MSVC v143 - VS 2022 C++ x64/x86 build t...
    ☑ C++ ATL for latest v143 build tools (x86 &...
    ☑ Security Issue Analysis
    ☑ C++ Build Insights
    ☑ Just-In-Time debugger
    ☑ C++ profiling tools
    ☑ C++ CMake tools for Windows
    ☑ Test Adapter for Boost.Test
    ☑ Test Adapter for Google Test
    ☑ Live Share
    ☑ IntelliCode
    ☑ C++ AddressSanitizer

Location
C:\Program Files\Microsoft Visual Studio\2022\Community Change...

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 10 GB

Install while downloading ▾ Install

Remove out-of-support components

ENG 02:13 PM

Configuration: Active(Debug) ▾   Platform: Active(x64) ▾   Configuration Manager...

▲ Configuration Properties
  General
  Advanced
  Debugging
  VC++ Directories
  ▲ C/C++
    General
    Optimization
    Preprocessor
    Code Generation
    Language
    Precompiled Headers
    Output Files
    Browse Information
    External Includes
    Advanced
    All Options
    Command Line
  ▲ Linker
    General
    Input

Additional Include Directories	$(MSMPI_INC);$(MSMPI_INC)\x64
Additional #using Directories	
Additional BMI Directories	
Additional Module Dependencies	
Additional Header Unit Dependencie	
Scan Sources for Module Dependenc	No
Translate Includes to Imports	No
Debug Information Format	Program Database for Edit And Continue (/ZI)
Support Just My Code Debugging	Yes (/JMC)
Common Language RunTime Suppor	
Consume Windows Runtime Extensio	
Suppress Startup Banner	Yes (/nologo)
Warning Level	Level3 (/W3)
Treat Warnings As Errors	No (/WX-)
Warning Version	
Diagnostics Format	Column Info (/diagnostics:column)
SDL checks	Yes (/sdl)
Multi-processor Compilation	
Enable Address Sanitizer	No
Enable Fuzzer Support (Experimental	No

Additional Include Directories
Specifies one or more directories to add to the include path. Separate with ';' if more than one. (/I[path])

OK   Cancel   Apply

Configuration: Active(Debug)   **Platform:** Active(x64)   Configuration Manager...

Precompiled Headers	
Output Files	
Browse Information	
External Includes	
Advanced	
All Options	
Command Line	
▲ Linker	
General	
Input	
Manifest File	
Debugging	
System	
Optimization	
Embedded IDL	
Windows Metadata	
Advanced	
All Options	
Command Line	
▷ Manifest Tool	
▷ XML Document Generator	

Setting	Value
Output File	$(OutDir)$(TargetName)$(TargetExt)
Show Progress	Not Set
Version	
Enable Incremental Linking	Yes (/INCREMENTAL)
Incremental Link Database File	$(IntDir)$(TargetName).ilk
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	**$(MSMPI_LIB64)**
Link Library Dependencies	Yes
Use Library Dependency Inputs	No
Link Status	
Prevent Dll Binding	
Treat Linker Warning As Errors	
Force File Output	
Create Hot Patchable Image	
Specify Section Attributes	

Output File
The /OUT option overrides the default name and location of the program that the linker creates.

         OK   Cancel   Apply

Configuration: Active(Debug)   **Platform:** Active(x64)   Configuration Manager...

Precompiled Headers	
Output Files	
Browse Information	
External Includes	
Advanced	
All Options	
Command Line	
▲ Linker	
General	
Input	
Manifest File	
Debugging	
System	
Optimization	
Embedded IDL	
Windows Metadata	
Advanced	
All Options	
Command Line	
▷ Manifest Tool	
▷ XML Document Generator	

Setting	Value
Additional Dependencies	**msmpi.lib;%(AdditionalDependencies)**
Ignore All Default Libraries	
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the link command line. [i.e. kernel32.lib]
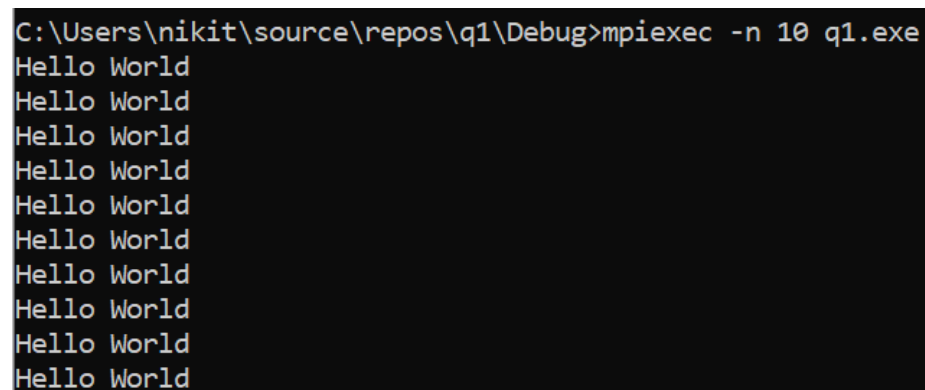
         OK   Cancel   Apply

Problem Statement 1:

Implement a simple hello world program by setting number of processes equal to 10

```cpp
#include <iostream>
#include <mpi.h>
using namespace std;
void main(int* argc, char* argv[]) {
        MPI_Init(&__argc, &__argv);
        cout << "Hello World!!" << endl;
        MPI_Finalize();
}
```

Screenshots:

```
C:\Users\nikit\source\repos\q1\Debug>mpiexec -n 10 q1.exe
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
```

Problem Statement 2:

Implement a program to display rank and communicator group of five processes

```cpp
#include <iostream>
#include <mpi.h>
int main(int argc, char** argv)
{
    MPI_Init(&argc, &argv);
    int rank, size;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank); // Get the rank of the current process
    MPI_Comm_size(MPI_COMM_WORLD, &size); // Get the total number of processes
    if (size < 5)
    {
        std::cerr << "This program requires at least 5 processes." << std::endl;
        MPI_Finalize();
        return 1;
    }
    MPI_Comm group_comm;
    int group_rank;
    int group_size;
    if (rank < 5)
    {
        MPI_Comm_split(MPI_COMM_WORLD, 0, rank, &group_comm);
        MPI_Comm_rank(group_comm, &group_rank);
        MPI_Comm_size(group_comm, &group_size);
        std::cout<<"Original rank "<<rank<<" of " <<size<< " in communicator of 5
processes, new rank " <<group_rank<<" of "<<group_size<<std::endl;

        MPI_Comm_free(&group_comm);
    }
    MPI_Finalize();
    return 0;
}
```

Screenshots:

```
C:\Users\nikit\source\repos\q1\Debug>mpiexec -n 5 q1.exe
Original rank 1 of 5 in communicator of 5 processes, new rank 1 of 5
Original rank 2 of 5 in communicator of 5 processes, new rank 2 of 5
Original rank 4 of 5 in communicator of 5 processes, new rank 4 of 5
Original rank 0 of 5 in communicator of 5 processes, new rank 0 of 5
Original rank 3 of 5 in communicator of 5 processes, new rank 3 of 5
```