Q1. Explain generic method in C++ Programming?
Ans:
C++ Templates, Templates are the foundation of generic programming. Which involves writing code in a way that is independent of any particular type.

A template is a blue print or formula for creating a generic classes or functions

========

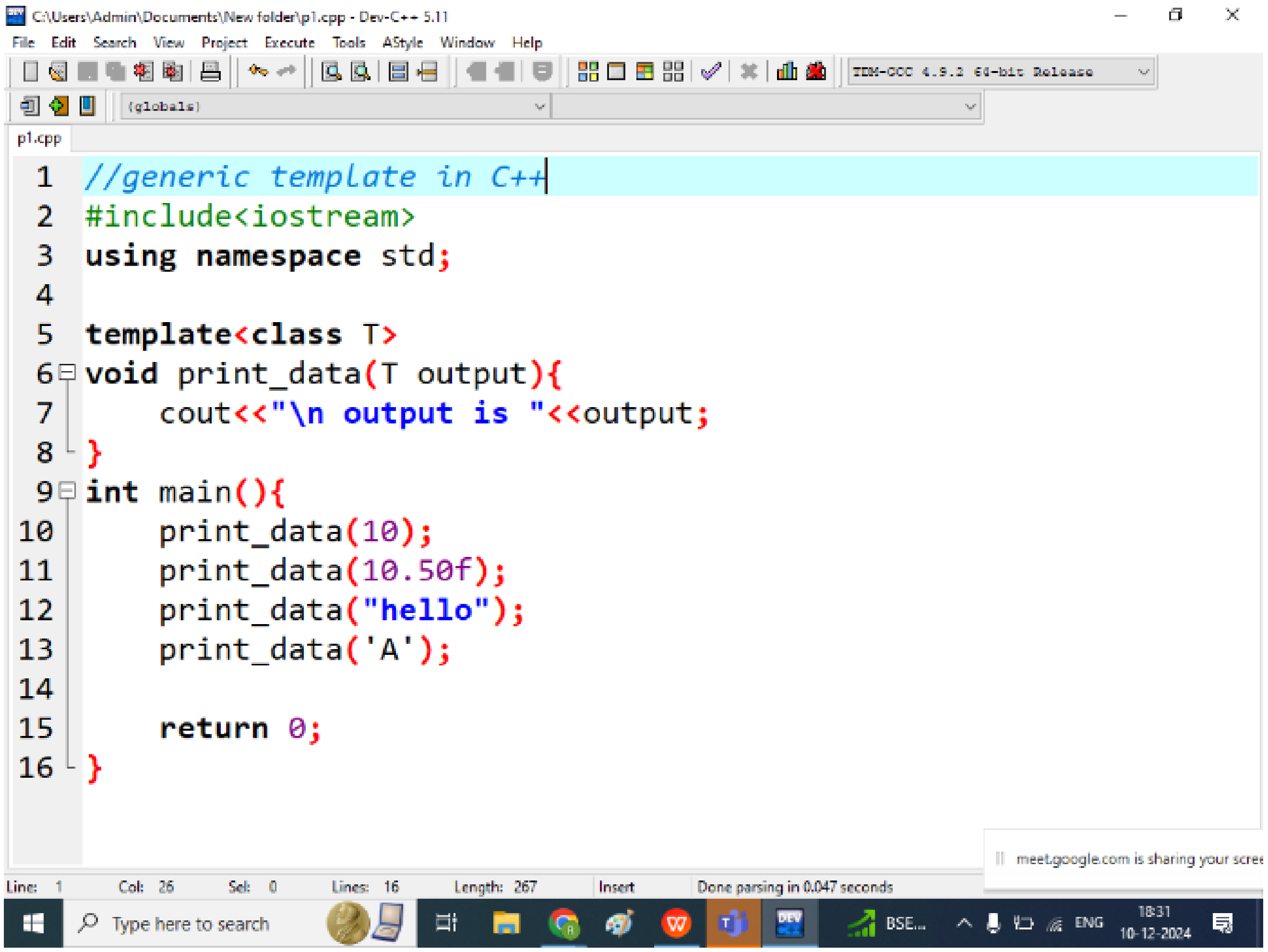
```
#include<iostream>
using namespace std;

void print_data(int x){
cout<<"\n int type value print : "<<x;
}
void print_data(float x){
cout<<"\n float type value print : "<<x;
}
void print_data(string x){
```

```
cout<<"\n string type value print : "<<x;
}
void print_data(char x){
cout<<"\n char type value print : "<<x;
}
int main(){
print_data(10);
print_data(10.0f);
print_data("hello");
print_data('A');




return 0;
}
```


C:\Users\Admin\Documents\New folder\p1.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
p1.cpp
Line: 1 Col: 26 Sel: 0 Lines: 16 Length: 267 Insert Done parsing in 0.047 seconds
meet.google.com is sharing your scree
Type here to search
BSE...
ENG
18:31
10-12-2024

```
//generic template in C++
#include<iostream>
using namespace std;

template<class T>
void print_data(T output){
    cout<<"\n output is "<<output;
}
int main(){
    print_data(10);
    print_data(10.50f);
    print_data("hello");
    print_data('A');

    return 0;
}
```

Q2. Write a c++ program to compare two different type values using template(generic function)

Example: Without generic and template
//generic template in C++
#include<iostream>
using namespace std;

template<class T>

```
int compare(int n1,int n2);
int compare(int n1,int n2){
return n1>n2?n1:n2;
}
float compare(float n1,float n2){
return n1>n2?n1:n2;
}
string compare(string n1,string n2){
return n1>n2?n1:n2;
}
char compare(char n1,char n2){
return n1>n2?n1:n2;
}
int main(){
int i1,i2;
i1=10;
i2=20;

cout<<"\n Largest Number in integer:
"<<compare(i1,i2);
float f1,f2;
f1=1.0f;
f2=1.1f;
cout<<"\n Largest Number in float:
"<<compare(f1,f2);
```

```
string s1,s2;
s1="HELLO";
s2="hello";
cout<<"\n Largest String :  "<<compare(s1,s2);
char c1='D';
char c2='d';
cout<<"\n Largest Characcter  :
"<<compare(c1,c2);
return 0;
}
//generic template in C++
#include<iostream>
using namespace std;

template<class T>
class Test{
public:
T obj;
//Member Data
Test(T x){
obj=x;
}
void showData(){
cout<<" Object is : "<<obj;
}
```

```
};

int main(){
Test<int> t1(10);
t1.showData();
Test<float> t2(10.5f);
t2.showData();
return 0;
}
```