**TEMPLATES:**

**10.1) Develop a C++ Program illustrating function template**

**Program:**

#include<iostream>

using namespace std;

template<typename T>

void change(T &x,T &y)

{

T temp;

temp=x;

x=y;

y=temp;

}

int main()

{

int a=6,b=10;

float x=10.2,y=12.4;

cout<<"before swapping is a &b:"<<a<<"\t"<<b<<endl;

swap(a,b);

cout<<"after swapping is a &b:"<<a<<"\t"<<b<<endl;

cout<<"before swapping is x &y:"<<x<<"\t"<<y<<endl;

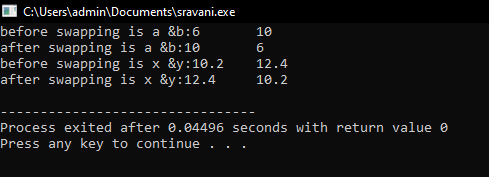
swap(x,y);

cout<<"after swapping is x &y:"<<x<<"\t"<<y<<endl;

return 0;

}

**Output:**



**10.2) Develop a C++ Program illustrating template class.**

**Program:**

#include<iostream>

using namespace std;

template<class T>

class sample

{

private:

T x;

public:

void get()

{

cout<<"enter x value:";

cin>>x;

}

void cube()

{

cout<<x\*x\*x;

}

};

int main()

{

sample<int>s;

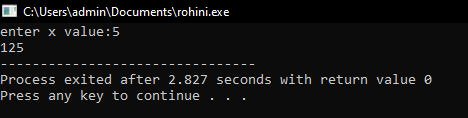
s.get();

s.cube();

return 0;

}

**Output:**



**10.3) Develop a C++ program to illustrate class templates with multiple parameters.**

**Program:**

#include<iostream>

using namespace std;

template<class T1,class T2>

class sample

{

private:

T1 a;

T2 b;

public:

void getdata()

{

cout<<"enter a,b values:";

cin>>a>>b;

}

void display()

{

cout<<"displaying values:"<<endl;

cout<<"a="<<a<<endl;

cout<<"b="<<b<<endl;

}

};

int main()

{

sample<int,int>s1;

sample<int,char>s2;

sample<int,float>s3;

cout<<"two integers data"<<endl;

s1.getdata();

s1.display();

cout<<"integer and character data"<<endl;

s2.getdata();

s2.display();

cout<<"integer and float data"<<endl;

s3.getdata();

s3.display();

return 0;

}

**Output:**

