**Practical : 1**

**Aim :** Write a "Garage" class that has a "Car" that is having troubles with its "Motor". Use a function-level try block in the "Garage" class constructor to catch an exception (thrown from the "Motor" class) when its "Car" object is initialized. Throw a different exception from the body of the "Garage" constructor s handler and catch it in main( ).

**Program :**

#include <iostream>

using namespace std;

class A

{

public:

void getdata(){

int a,b,c;

cout<<"enter 1st number :";

cin>>a;

cout<<"enter 1st number :";

cin>>b;

try

{

if(b==0)

{

throw 12;

}

else

{

c = a/b;

cout<<c;

}

}

catch(float n)

{

cout<<"not possible";

}

catch(double n)

{

cout<<"Error";

}

catch(...)

{

cout<<"genral exeption";

}

}

};

int main()

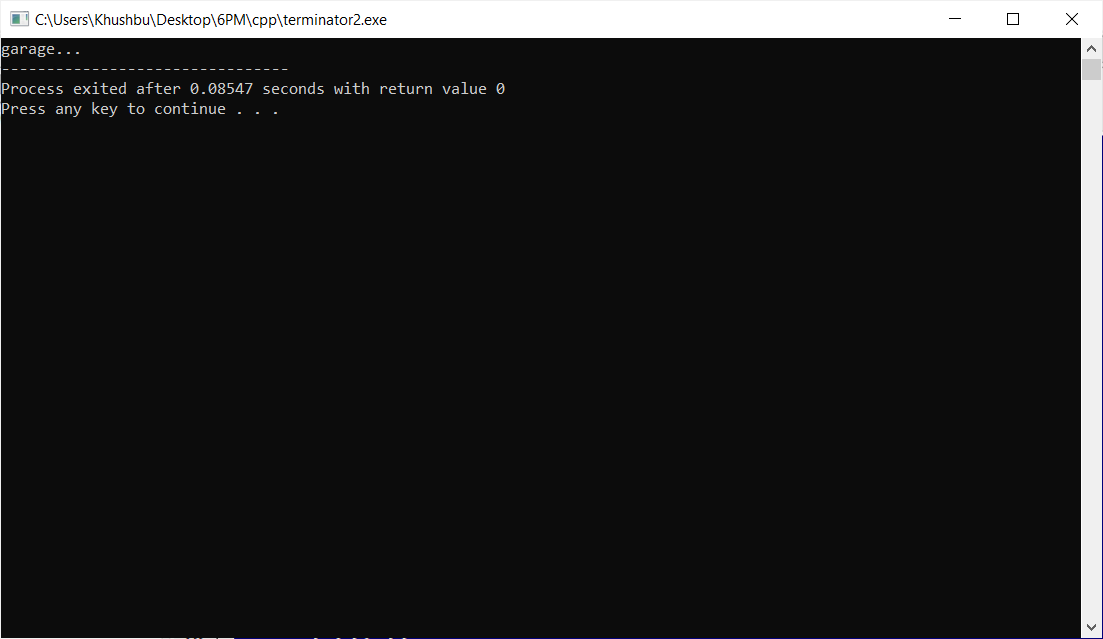
{

A a;

a.getdata();

}

**Output :**

****

**Practical : 2**

**Aim :** WAP to create a class which illustrate the concept of handling all types of exceptions using general exception.

**Program :**

#include <iostream>

using namespace std;

class A

{

public:

void getdata(){

int a,b,c;

cout<<"enter 1st number :";

cin>>a;

cout<<"enter 1st number :";

cin>>b;

try

{

if(b==0)

{

throw 12;

}

else

{

c = a/b;

cout<<c;

}

}

catch(float n)

{

cout<<"not possible";

}

catch(double n)

{

cout<<"Error";

}

catch(...)

{

cout<<"genral exeption";

}

}

};

int main()

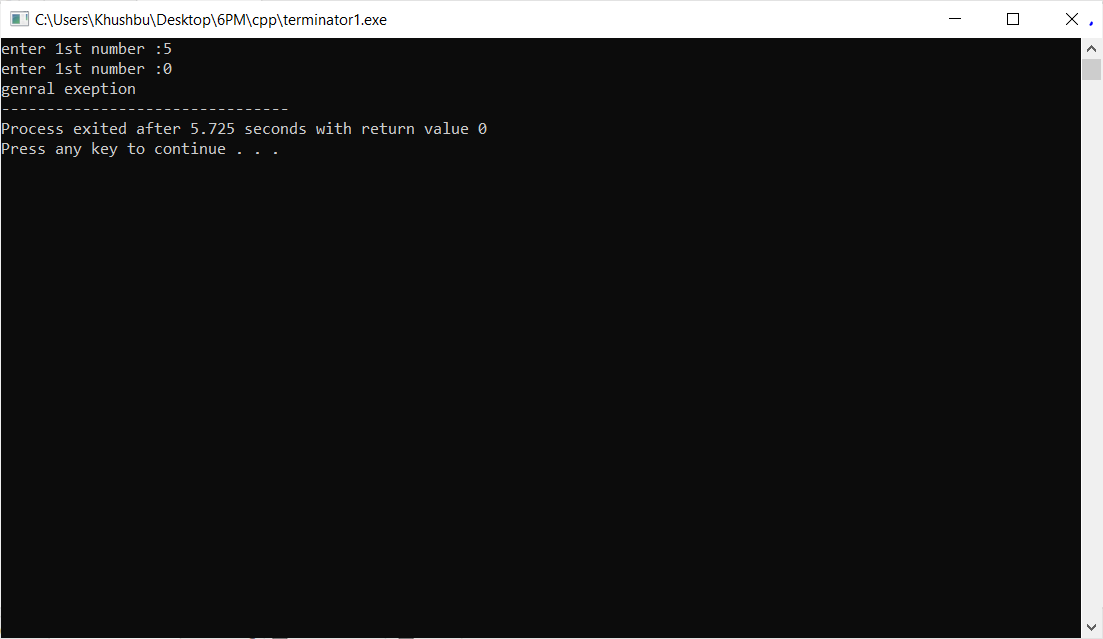
{

A a;

a.getdata();

}

**Output :**

****