**/\* 34. Write a function that handles 'divide by zero' and 'divide by negative number' internally,allowing the user to correct the error,but throws a bad operator exception**

**to the calling function \*/**

#include<iostream>

using namespace std;

class Error

{

public:

virtual void print()

{

cout<<"\*\* Error: type error\n";

}

};

class arithmetic

{

public:

virtual void print()

{

cout<<"\*\* Error: arithmetic error\n";

}

};

class divbyzero: public arithmetic

{

public:

virtual void print()

{

cout<<"\*\* Error: 100 divisor 0\n";

}

};

class divbyneg :public arithmetic

{

public:

virtual void print()

{

cout<<"\*\* Error 101 negative divisor\n";

}

};

class badoperator: public arithmetic

{

public:

virtual void print()

{

cout<<"\*\* Error 102 invalid operator\n";

}

};

double math(char oper,double data1,double data2)

{

double result;

switch(oper)

{

case'+':

result=data1+data2;

break;

case'-':

result=data1-data2;

break;

case'\*':

result=data1\*data2;

break;

case'/':

if(data2==0)

throw divbyzero();

if(data2<0)

throw divbyneg();

result=data1/data2;

break;

default:

throw badoperator();

break;

}

return result;

}

int main()

{

double data1,data2;

cout<<"\nEnter the first data:- ";

cin>>data1;

cout<<"\nEnter the second data:- ";

cin>>data2;

char oper;

cout<<"\nEnter the operator:- ";

cin>>oper;

try

{

double r=math(oper,data1,data2);

cout<<"Result: "<<r<<endl;

}

catch(Error &error)

{

error.print();

}

system("pause");

return 0;

}

/\* Output :-







