بسم الله الرحمن الرحيم

Object Oriented Programming

البرمجة الشيئية

عدد الساعات: ٢ نظري + ٢ عملي

المتطلبات السابقة: ٢١١حسب

أ/المادة: م. نجلاء حسن

Lecture 12

Exceptions &Handling الاستثناءات ومعالجة الأخطاء

أنواع الاخطاء

1- Syntax error
2- Runtime error
3- Logical error

RUNTIME ERRORS

هي الأخطاء التي تظهر أثناء تشغيل البرنامج مثل:

- الكتابة علي قطاع تالف bad sector
- الكتابة علي ملف لم يعد موجود
- عدم القدرة على فتح أحد الملفات
- اضافة شي لقاعدة بيانات لا تملك الحق بذلك
- القسمة علي صفر

اهمية الاستثناءات

توفر علي المبرمج كتابة الكثير من الاكواد لمعالجة اخطاء قد تظهر في وقت التنفيذ.

```
read_file () // دالة للقراءة من الملف
{
Open the file
Determine the size
Allocate enough memory
Read the file
Close the file
}
```

الخوارزم لم يعالج أي خطاء من الاخطاء المتوقعة حدوثها مثل ؟؟

١. عدم القدره علي فتح الملف لنقله او حذفه او تغير اسمه ..
٢. عدم القدرة علي تحديد حجم الملف..
٣. لا يوجد مكان كافي بالذاكرة لاحتواء الملف..
٤. فشل القراءة من الملف
٥. فشل بإغلاق الملف..

نعدل الخوارزم لمعالجة الاخطاء

```
read_file ()
{
   try
    {
Open the file
Determine the size
Allocate enough memory
Read the file
Close the file
}
Catch(Open the file failed)
{do some thing}
catch(size determination failed)
{do some thing}
catch(memory allocation failed)
{do some thing}
Catch(file closed filed)
{do some thing}
}
```

- تنفيذ الكود بداخل try اذا وجد خطأ ما سيقوم بمعالجة الخطأ من خلال الكود الموجود بال catch
- هذه الطريقة لمعالجة الاخطاء تسمي Exception Handling

أدوات معالجة الاستثناءات

معالجة الاستثناءات من خلال الكلمات :

try , catch , throw

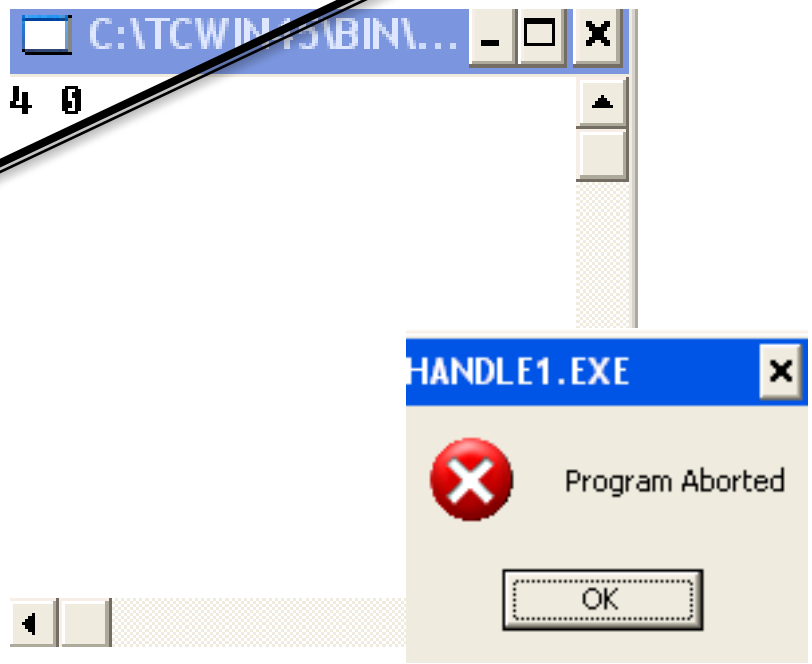
استخدام TRY..CATCH

```
Try
{
…
…
Throw Exception()
…
…
}
Catch(Exception e)
{
…
…
}
```

الخطأ الواقع يتم ترحيله لأول catch اما يعالجه او يتم ترحيله الي ال catch التالية وهكذا .. فان لم يجد catch مناسبه فسيتم قطع التفيذ للبرنامج.

استخدام throw بدون استثناء

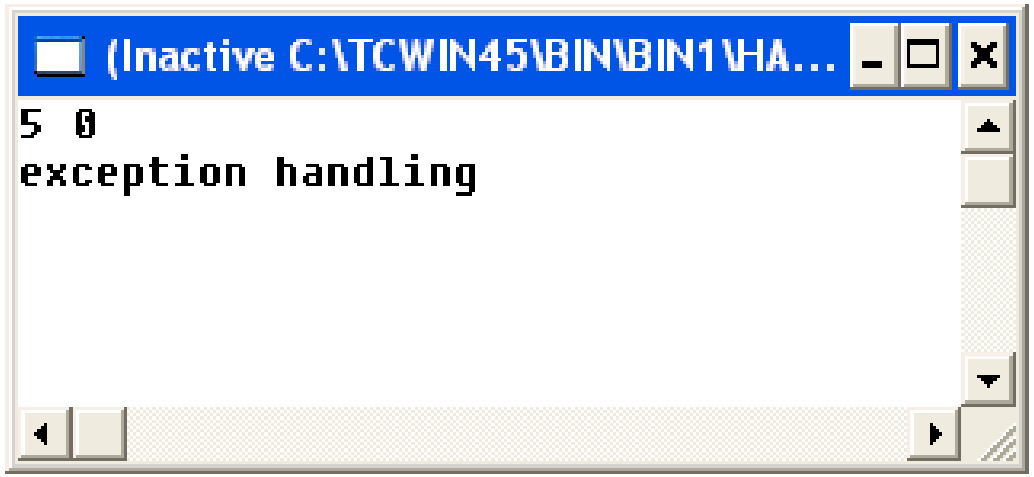
```
#include<iostream.h>
//#include<string.h>
void main(){
int x,y;
cin>>x>>y;
try{
if(y==0)
throw;

cout<<x/y;}
catch(...){}
}
```



يتوقف تنفيذ البرنامج ويعطي الرسالة اعلاه

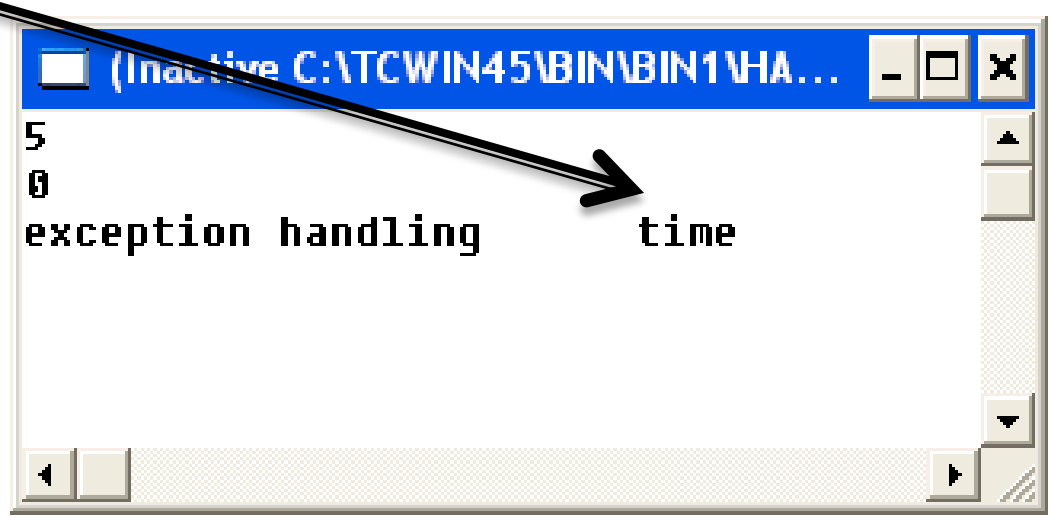
استخدام throw مع استثناء

```
#include<iostream.h>
//#include<string.h>
void main(){
int x,y;
cin>>x>>y;
try{
if(y==0)
throw "time";

cout<<x/y;}
catch(...){cout<<"exception handling";}
}
```



لن يتوقف البرنامج ويعطي الرسالة الموضحة اعلاة (الموجودة فى catch)

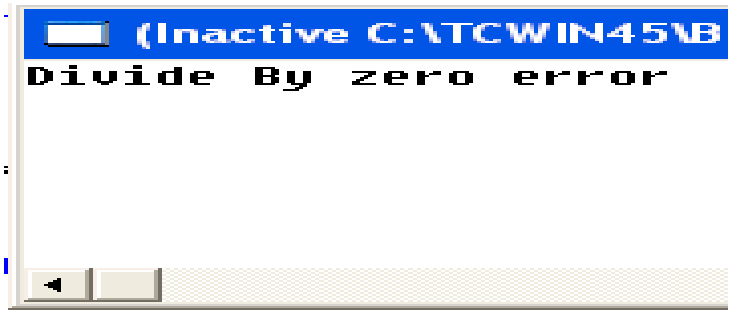
```
#include<iostream.h>
//#include<string.h>
void main(){
int x,y;
cin>>x>>y;
try{
if(y==0)
throw "time" ;

cout<<x/y;}
catch(char *x){cout<<"exception handling"<<"\t"<<x;}
}
```



```
#include<iostream.h>
const    int   DivideByzero=20;
double   divide(double x, double y)
{
            if(y==0)
             {
                        throw DivideByzero;
             }
                    return x/y;
}

void main()
{
  try
  {
             divide(10,0);
  }
     catch(int i)
      {
           if(i== DivideByzero)
             {

                 cerr<<"Divide By zero error";
             }
      }
 }
```

- مثال: القسمة على صفر

امر throw :يقوم برمي الاستثناء ليلتقطه المعالج (جمل catch) لفحصه واجراء اللازم (معالجة الخطأ)

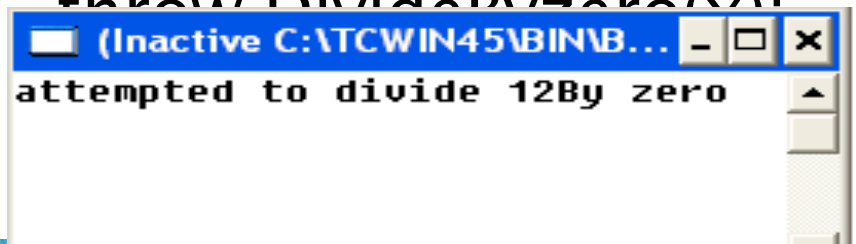


```
#include<iostream.h>
int main()
{
      int   studentAge;
    try {
                  cout<< "student Age:";
                  cin>> studentAge;
                  if(studentAge <=0)
                        throw "positive number reguired";
                  cout<<"\n student Age:
"<<studentAge<<"\n\n";
        }
                catch (const char * Message)
                {
                  cout<<"Error: "<< Message;
                 }
                 cout<<"\n";
      return 0;
 }
```

ما هي المخرجات اذا تم ادخال 0

```
#include<iostream.h>
class DivideByzero
{
  public:
     double d;
     divideByzero(double x);
 };

DivideByzero::DivideByzero(dou
ble x):d(x) {}
 void divide(int x ,int y)
{
      if(y==0)
        {
          throw DivideByzero(y);
}
```

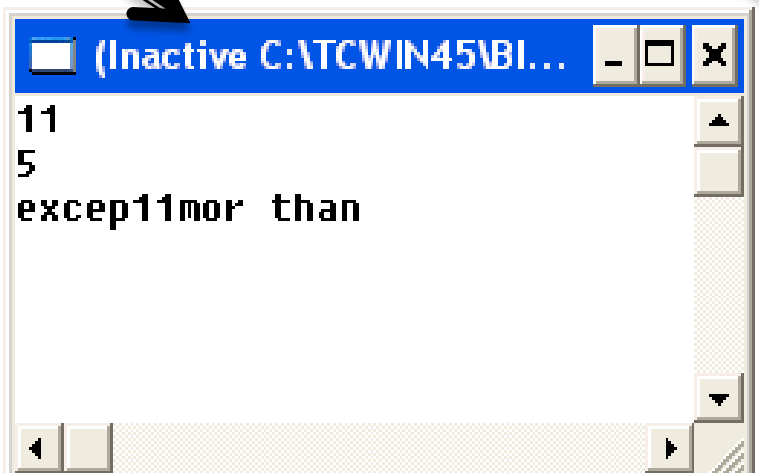
```
void main()
{
  try
  {
      divide(12,0);
   }

catch(DivideByzero divzero
)
   {
  cerr<<"attempted to
divide” << divzero.d<<"By
zero";
   }
 }
```

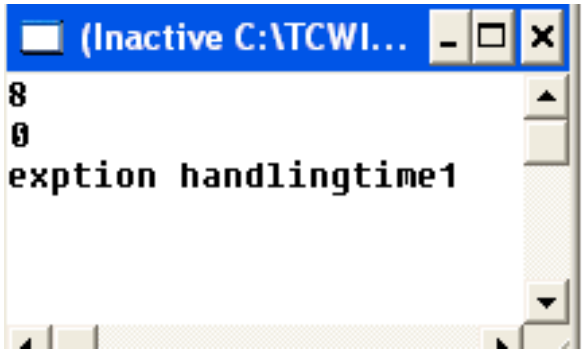


catching Multiple Exceptions

```
try{
        code to try
}
catch(Arg1)
{
    one Exception
}
catch(Arg2)
{
    Another Exception
}
```

```
#include<iostream.h>
void main(){
int x,y;
cin>>x>>y;
try{
if(y==0)
throw "time1";
if (x>10 )
   throw x;
cout<<x/y;}
catch(char * x) {cout<<"exption handling"<<x<<endl;}
catch (int x){cout<<"excep"<<x<<"mor than";}
}
```





في حال توافق نمط البيانات الذي تلقيه الكلمة throw مع أي كتلة catch فان التنفيذ ينتقل الى هذه الكتلة