University of Hertfordshire

7COM1025
Programming for Software Engineers
Lecture 23

Dr. Renato C. de Amorim

THE ONE HUNDRED UNDER FIFTY

UNIONS

They allow a single portion of the memory to be accessed by different variables (even with different data types).

```
#include <iostream>
using namespace std;
union DataValue{
    int v_int;
    float v_float;
    double v_double;
};
int main () {
    DataValue MyValue;
    MyValue.v_double = 1.12345;
    cout<<"Value: "<<MyValue.v_double<<endl;
    MyValue.v_float = 1.12f;
    cout<<"Value: "<<MyValue.v_float<<endl;
    MyValue.v_int = 1;
    cout<<"Value: "<<MyValue.v_int<<endl;
    return 0;
}
```

ENUMERATED TYPES

```
#include <iostream>
#include<string>
using namespace std;
enum ValidColours{
    Black,
    White,
    Green,
    Blue,
};
void PrintColour(string Str, ValidColours ChosenColour)
{
    switch(ChosenColour){
    case White:
        cout<<Str<<endl;
    }
}
int main () {
    ValidColours MyColour;
    MyColour = White;
    PrintColour("Test", MyColour);
    return 0;
}
```

In C++ 11 you can also choose the underling type.

```
#include <iostream>
#include<string>
using namespace std;
enum ValidColours: char{
    Black,
    White,
    Green,
    Blue,
};
```

STRUCTS

Structs have default of public, while classes have a default of private.
You can have a template class, but not a template struct.
Both can use inheritance.
Classes cannot be used when interfacing with C because C does not have classes (unlike C++).

```
#include<iostream>
using namespace std;
struct Vec{
float x, y, z;
float length() {return x+y+z;}
Vec() {x=y=z=0.0F;}};
int main() {
    Vec MyVector;
    cout<<"Enter x, y, and z"<<endl;
    cin>>MyVector.x>>MyVector.y>>MyVector.z;
    cout<<MyVector.length();
    return 0;
}
```

PROBLEM 23.1

Structures are sometimes used instead of classes for Plain Old Data (POD). These don't have constructors, virtual functions or inheritance.

Colours are normally represents using either RGBA or an unsigned int value. Making use of unions and structs create a C++ representation for Colours.

Now Write a class Colour. This class should have the structure you created, plus some way for the user to set the name of the colour.

PREPROCESSOR DIRECTIVES

The pre-processor allows for modifications of the source code before this is send to the compiler.

#define
define constants

```
#include<iostream>
#define API_MAJOR 1
#define API_MINOR 2
#define API_PATCH 3
#define getmax(a,b) ((a)>(b)?(a):(b))
int main()
{
   std::cout<<"Version: "<<API_MAJOR<<'.'<<API_MINOR<<'.'<<API_PATCH<<std::endl;
   std::cout<<"The max number is "<<getmax(5,8);
   return 0;
}
```

- Think of it as an “automatic search and replace”.
- They don't have any scope
- There is no type checking.

PREPROCESSOR DIRECTIVES

```
#include<iostream>
#ifndef getmax
#define getmax(a,b) ((a)>(b)?(a):(b))
#endif
using namespace std;
int main()
{
   cout<<"The max number is "<<getmax(5,8);
   return 0;
}
```

One of the major uses of #ifndef is to avoid including the same header twice.

PREPROCESSOR DIRECTIVES

```
#include<iostream>
#ifndef getmax
#error getmax is not defined!
#endif
using namespace std;
int main()
{
   std::cout<<"The max number is "<<getmax(5,8);
   return 0;
}
```

PREPROCESSOR DIRECTIVES

#line

In the case of an error, the compiler normally throws a line number and an error message. You can change these if you want.

```
#include <iostream>
using namespace std;
int main()
{
  #line 200
  This is a compiler error at line 200;
  return 0;
}
```

STANDARD MACROS

```
#include <iostream>
using namespace std;
int main()
{
  cout << "This is the line number " << __LINE__;
  cout << " of file " << __FILE__ << ".\n";
  cout << "Its compilation began " << __DATE__;
  cout << " at " << __TIME__ << ".\n";
  cout << "The compiler gives a __cplusplus value of " << __cplusplus;
  return 0;
}
```

PROBLEM 23.2

Improve your previous exception class so that it also contains information about the name of the file throwing the exception, and the date and time of compilation.