CSC215

Math and Computer Science

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

Lets do a Temperature ADT

- If I had a single temperature, what can I do to it?

Create

Destroy

What I Came Up With

Create

Destroy

Set the temperature using Fahrenheit, Celsius, or Kelvin value

Retrieve the Fahrenheit, Celsius, or Kelvin temperature

Output the Fahrenheit, Celsius, or Kelvin temperature to any ostream.

Swap two temperature values

Compare two temperature values

Functional Abstraction

Constructors

Create object with no temperature supplied

in: nothing out: nothing

Create object with a Fahrenheit temperature

in: Fahrenheit temp out: nothing

Create an object using an existing temperature class

in: temperature class out: nothing

Destructors

in: nothing out: nothing

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Set Temperature Functions

Set Fahrenheit

in: a Fahrenheit Temperature out: nothing

Set Celsius

in: a Celsius Temperature out: nothing

Set Kelvin

in: a kelvin Temperature out: nothing

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Get Temperature Functions

Get Fahrenheit

in: nothing out: equivalent Fahrenheit temperature

Get Celsius

in: nothing out: equivalent Celsius temperature

Get Kelvin

in: nothing out: equivalent Kelvin temperature

Problem: how will I know what unit the temperature is in?

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Rethink Set Temperature Functions

Set Fahrenheit

in: a Fahrenheit Temperature out: nothing

Set Celsius – will convert the Celsius temperature passed in to F

in: a Celsius Temperature out: nothing

Set Kelvin – will convert the Kelvin temperature passed in to F

in: a Kelvin Temperature out: nothing

Problem Solved: Data in object always stored as Fahrenheit.

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Output Temperatures

Output Fahrenheit Temperature

in: output stream output: Fahrenheit temperature

Output Celsius Temperature

in: output stream output: Celsius temperature

Output Kelvin Temperature

in: output stream output: Kelvin temperature

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Compare Function

Compare

in: a temperature class to compare it to

out: the difference between the two temperatures

Swap Function

Swap

in: a temperature class

out: nothing

Writing the Class Definition

```
#ifndef __TEMPERATURE__H__
#define __TEMPERATURE__H__
class Temperature
{   public:
        Temperature();
        Temperature( double fahrenheitTemp );
        Temperature( Temperature &t );
        ~Temperature();
```

Note: see slide 4 for information

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Set the Temperature Functions

```
#ifndef __TEMPERATURE__H__
#define __TEMPERATURE__H__
class Temperature
{   public:
        Temperature();
        Temperature( double fahrenTemp );
        Temperature( Temperature &t );
        ~Temperature();
        void setFahrenheit(double farTemp );
        void setCelsius( double celTemp );
        void setKelvin( double kelTemp );
```

Note: see slide 7 for information

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Get the Temperature Functions

```
#ifndef __TEMPERATURE__H__
#define __TEMPERATURE__H__
class Temperature
{   public:
        double getFahrenheit( );
        double getCelsius( );
        double getKelvin( );
```

Note: See slide 6 for Information

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Output Specific Temperatures

```
#ifndef __TEMPERATURE__H__
#define __TEMPERATURE__H__
class Temperature
{   public:
        void outputFahrenheit( ostream &out );
        void outputCelsius( ostream &out );
        void outputKelvin( ostream &out );
```

Note: see slide 8 for information

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Compare and Swap Functions

```
#ifndef __TEMPERATURE__H__
#define __TEMPERATURE__H__
class Temperature
{   public:
        double compare( Temperature &T );
        void swap( Temperature &T );
```

Note: See Slides 9 & 10 for information

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Add the Storage for the Data

```
#ifndef __TEMPERATURE__H__
#define __TEMPERATURE__H__
class Temperature
{   public:
        Temperature();
        Temperature( double fahrenTemp );
        Temperature( Temperature &t );
        ~Temperature();

    private:
        double theTemp;
};
#endif
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

The temperature.h file

```
#ifndef __TEMPERATURE__H__
#define __TEMPERATURE__H__
class Temperature
{   public:
        Temperature();
        Temperature( double fahrenTemp );
        Temperature( Temperature &t );
        ~Temperature();
        void setFahrenheit(double farTemp );
        void setCelsius( double celTemp );
        void setKelvin( double kelTemp );
        double getFahrenheit( );
        double getCelsius( );
        double getKelvin( );
        void outputFahrenheit( ostream &out );
        void outputCelsius( ostream &out );
        void outputKelvin( ostream &out );

        double compare(Temperature &T );
        void swap( Temperature &T );

    private:
        double theTemp;
};

#endif
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Writing the source code: (.cpp) file

```
#include "temperature.h"
Temperature::Temperature()
{
    theTemp = 0.0;
}
Temperature::Temperature(
                        double ftemp )
{
    theTemp = ftemp;
}
```

```
Temperature::Temperature( Temperature &t )
{
    theTemp = t.theTemp;
}

Temperature::~Temperature()
{

}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Set Functions

```
void Temperature::setFahrenheit( double farTemp )
{
    theTemp = farTemp;
}

void Temperature::setCelsius( double celTemp ) // remember to convert to F
{
    theTemp = 9.0 / 5.0 * celTemp + 32.0;
}

void Temperature::setKelvin( double kelTemp ) // remember to convert to F
{
    theTemp = ( ( kelTemp - 273 ) * 9.0 / 5.0 ) + 32.0;
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Get Functions

```
double Temperature::getFahrenheit( )
{
    return theTemp;
}
double Temperature::getCelsius( ) // convert F to C
{   double tval;
    tval = ( theTemp - 32.0 ) * 5.0 / 9.0;
    return tval;
}
double Temperature::getKelvin( ) // convert F to K
{   double tval;
    tval = ( 5.0 / 9.0 * (theTemp - 32.0) + 273 );
    return tval;
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Output Functions

```
void Temperature::outputFahrenheit( ostream &out )
{   unsigned char degree = 248;
    out << theTemp << degree << "F";
}
void Temperature::outputCelsius( ostream &out )
{   unsigned char degree = 248;
    out << getCelsius() << degree << "C";
}
void Temperature::outputKelvin( ostream &out )
{   unsigned char degree = 248;
    out << getKelvin() << degree << "K";
}
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Compare Function

```
double Temperature::compare(Temperature &T )
{
    return theTemp - T.theTemp;
}
```

Notes: if the value returned is 0, they are equal

if the value returned is <0, a < T

if the value returned is >0, a > T

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

The Swap Function

```
void Temperature::swap( Temperature &T )
{
    double tval;

    tval = T.theTemp;
    T.theTemp = theTemp;
    theTemp = tval;
}
```

For Your Own Exercise

- Design an ADT for a date
- Design an ADT for a time

- Try and write the code for this.