Java Programming

Arthur Hoskey, Ph.D.
Farmingdale State College
Computer Systems Department

- Generics

Lesson Topics

- Generics are like templates in C++ and generics in C#.
- Instead of hardcoding a type in you can let the type change according to your needs.
- Create classes and methods that are independent of contained types.
- No need to write multiple methods with the same functionality, just write one.

Generics

```
class Data
{
   private int x;

   public int GetX() {
         return x;
   }

   public void SetX(int newx) {
         x = newx;
   }
}
```

Data Class

Stores one piece of <u>integer</u> data.

Store One Specific Type of Data

- We have a class that can hold one piece of integer data.
- What if we wanted a class that could store one piece of string data?
- Could we use the class we just wrote?

Store One Specific Type of Data

```
class Data
{
   private String x;

   public String GetX() {
         return x;
   }

   public void SetX(String newx) {
         x = newx;
   }
}
```

Data Class

Stores one piece of string data.

Store One Specific Type of Data

- If we wanted to create a class that could hold boolean data, we would have to rewrite the data class, yet again.

Store One Specific Type of Data

```
class Data
{
   private boolean x;

   public boolean GetX() {
         return x;
   }

   public void SetX(boolean newx) {
         x = newx;
   }
}
```

Data Class

Stores one piece of bool data.

Store One Specific Type of Data

- There is a better way to do this.
- Use generics instead.
- Write it once but be able to store different types of data.

Generics

- Generics allow you to write a class or method that can be used with different data types.

- For example…

Generics

```
class Data<T>
{
    private T x;
    public T GetX()
    {
        return x;
    }

    public void SetX(T newx)
    {
        x = newx;
    }
}
```

T is a type parameter

Data Class

This version uses generics.

The type parameter (T in this case) is replaced with a value or reference type.

There is nothing special about T you can basically use any name for the type parameter.

Generics

```
public static void main(String[] args)
{
   Data<Integer> d;
   d = new Data<Integer>();

   d.SetX(10);

   System.out.println(d.GetX());

   Data<String> d2;
   d2 = new Data<String>();
   d2.SetX("Yanks");

   System.out.println(d2.GetX());
}
```

Need to pass in the data type as part of the variable declaration when using templates

IMPORTANT!!!

You MUST use the same data type for the generic field when creating the instance or there will be a compile error.

This code uses the generics version of Data.

Must pass in the data type to use for an instance of the class.

Generics

```
public static void main(String[] args)
{
   Data<Integer> d;
   d = new Data<String>();

   d.SetX(10);
   d.SetX("Yanks");
}
```

What is wrong with this code?

Generics

```
public static void main(String[] args)
{
   Data<Integer> d;
   d = new Data<String>();

   d.SetX(10);
   d.SetX("Yanks");
}
```

What is wrong with this code?

Incorrect type parameter used when creating the instance.

The d variable was declared as having int for the type but the call to new uses string.

You must use the same type parameter when calling new.

Generics

```
class Data<T> {
    private T x;
    private int i;

    public T GetX()  {
        return x;
    }

    public void SetX(T newx) {
        x = newx;
    }

    public int GetI()  {
        return i;
    }

    public void SetI(int newi) {
        i = newi;
    }
}
```

Generics

You are allowed to create member variables with normal data types even if you use generics in a class definition.

The type parameter does not affect a variable that has a normal value or reference type as the data type.

Generic Methods

You can use generics on individual methods (can also be static)

```
public class Main {

  static <T> void test(T data) {
      System.out.println(data);
  }

 public static void main(String[] args) {
      Main.<Integer>Test(100);
   }

}
```

Define the method with a generic

Test method call requires the data type

Calling a generic static method requires that we put the class name first

- **End of Slides**

End of Slides