C# Quiz
Second Year Project Course
B. Sc. course

Niels Hallenberg
IT University of Copenhagen

Why a C# Quiz

- Align C# lectures.
- Purpose of C# lectures to know *the basics* necessary for F#.
- We also want to be able to explain how LINQ works.
- The basics can be hard to comprehend (delegates, lambda’s, extension methods, type inference, anonymous objects, function types,…).
- There are 1000 details about C# - not important here – we can always pose questions about C# that we can’t answer – in that way C# is not the most beautiful language ever created.
“C# will keep growing and some day it may collapse under its own weight”

Anders Hejlsberg at the IT University 2007 (quoted from memory)

---

C# Quiz (CSharp_Quiz.sln)

Q1 The variable \texttt{a.x} below represents
a) an object of a class
b) an instance method
c) an instance variable
d) none of the above

```csharp
class Foo {
    int x = 100;
    public static void Main() {
        Foo f = new Foo();
        Console.WriteLine(a.x);
    }
}
```
C# Quiz

Q2 The statement below
   a) creates a Button control
   b) initializes a button control
   c) instantiates button control

Choose all that is true.

Private Button doSave = new Button();

C# Quiz

Q3 What is true about a Constructor?
   a) it is used to create and instantiate objects
   b) it must have the same name as the class it is declared within
   c) it maybe overloaded
   d) a constructor in a possible base class is always called
   e) a, b and c
   f) all of the above
C# Quiz

Q4 Does the program below compile?
   a) Yes
   b) No

```csharp
public class Top {
    public Top(int i) {
        Console.WriteLine("Top " + i);
    }
}
public class Buttom : Top {
    public Buttom() {
        Console.WriteLine("Buttom");
    }
}
class Program {
    static void Main(string[] args) {
        Buttom b = new Buttom();
    }
}
```

C# Quiz

Q5 A delegate defines
   a) a concept of passing methods as first class values
   b) a class that encapsulates methods
   c) a means of passing collections into methods
   d) a governmental representative

Choose all that is true above.
C# Quiz

Q6 Polymorphism occurs when a method in a child class
a) override a method in the parent class by extending the parameter list but maintains the same return type.
b) maintain the same method signature as the method in the parent class – however the implementation differs.
c) override a method in the parent class by maintaining the same parameter list but changes the return type.
d) a and c

C# Quiz

Q7 Method Hiding – what is the output?

a) A.foo B.foo A.foo
b) A.foo B.foo B.foo

The compiler gives a warning – why?

How to avoid the warning and maintain the same semantics?
C# Quiz

Same as before without warning from the compiler.

```csharp
public class A {
    public void foo() {
        Console.WriteLine("A.foo");
    }
}

public class B : A {
    public new void foo() {
        Console.WriteLine("B.foo");
    }
}

class Program {
    static void Main(string[] args) {
        A a = new A();
        B b = new B();
        a.foo();
        b.foo();
        ((A)b).foo();
    }
}
```

Q8 Method Overriding – what is the output?

a) A.foo B.foo A.foo
b) A.foo B.foo B.foo

This is what we normally expect and also known from Java.

When you put objects of different inherited types into a collection and apply a method on each of them, then Hiding and Overriding makes a huge difference – and is also confusing.

```csharp
public class A {
    public virtual void foo() {
        Console.WriteLine("A.foo");
    }
}

public class B : A {
    public override void foo() {
        Console.WriteLine("B.foo");
    }
}

class Program {
    static void Main(string[] args) {
        A a = new A();
        B b = new B();
        a.foo();
        b.foo();
        ((A)b).foo();
    }
}
```
C# Quiz

Q9 About abstract classes and methods. Does the program below compile?
   a) Yes
   b) No

```csharp
public abstract class AbstractTop {
    public abstract int foo(int i);
    public int foo2(int i) {
        Console.WriteLine("foo2: " + i);
        return i;
    }
}

public class NotAbstractTop : AbstractTop {
    public int foo(int i) {
        Console.WriteLine("foo: " + i);
        return i;
    }
}
```

C# Quiz

Q10 Can a sealed class be a base class?
   a) Yes
   b) No

Q11 What is true about an abstract class?
   a) may contain constructors
   b) may contain implemented methods
   c) may extend other classes
   d) may contain instance variables
   e) a and b and c
   f) all of above
**C# Quiz**

**Q12** Say you are coding a threaded GUI application based on Windows Forms. Can you safely update the value of a GUI control outside the main GUI thread?

a) Yes  
b) No

**Q13** What is true about interfaces in C#?

a) an interface can inherit from another interface  
b) an interface can inherit from many interfaces  
c) a class can implement one interface  
d) a class can implement many interfaces

Choose all that apply

---

**C# Quiz – IEnumerable,IEnumerator, Collections**

**Q14** For the code below to compile and run the minimum requirements to `xs` are:

a) `xs` implements `IEnumerable`  
b) `xs` implements `IEnumerator`  
c) `xs` implements both

```csharp
foreach (var i in xs)
    Console.WriteLine(i);
```

**Q15** What is common to the following collections:

- `System.Collection.ArrayList`
- `System.Collection.HashTable`
- `System.Collection.Queue`
- `System.Collection.SortedList`
- `System.Collection.Stack`

a) They get the best out of polymorphism, type safety and performance.  
b) No – polymorphism is obtained with almost no type safety and lots of inefficient boxing and un-boxing at runtime.
C# Quiz – Generics

Q16 Does the nonsense to the right compile?
   a) Yes
   b) No

```csharp
public interface IA { void fooA(); }
public interface IB { void fooB(); }
public class MyGeneric<T> where T : IA, IB {
    T x;
    public MyGeneric(T x) { this.x = x; }
    public void pp() { this.x.fooA();
                      this.x.fooB();
    }
}

public class AB : IA {
    public void fooA() { Console.WriteLine("fooA");}
    public void fooB() { Console.WriteLine("fooB");
}

class Program {
    static void Main(string[] args) {
        AB ab = new AB();
        var abInt = new MyGeneric<AB>(ab);
        abInt.pp();
    }
}
```

C# Quiz – Generics and Lambda

Q17 What is written on the console?
Q18 Can ListUtil be used on List with elements of any type?

```csharp
public static class ListUtil<T> {
    public static List<T> map(Func<T, T> fn, List<T> xs) {
        List<T> map2 = new List<T>();
        foreach (T x in xs) map2.Add(fn(x));
        return map2;
    }
    public static void apply(System.Action<T> fn, List<T> xs) {
        foreach (T x in xs) fn(x);
        return;
    }
}

var intList = new List<int>(new[] {1,2,3,4});
var intList2 = ListUtil<int>.map((x) => x+3, intList);
var intList3 = ListUtil<int>.map((x) => x*2, intList2);
ListUtil<int>.apply((x) => Console.WriteLine(x), intList3);
```
C# Quiz – Indexers and Operators

Q19 What is true about indexers?
   a) You can index on integer values
   b) You can index on string and integer values
   c) You can overload indexers
   d) You can have multidimensional indexers
   e) You can index on other values than string and integers
Choose all that is true.

Q20 What is true about operator overloading?
   a) Overloading works for unary operators
   b) Overloading works for binary operators
Choose all that is true.

C# Quiz – Type Inference

Q21 What is true about type inference?
   a) type inference works in for and foreach
   b) type inference works in method parameters
   c) type inference works for return types
   d) local variable declarations
   e) type inference works for const declarations
   f) type inference works for class fields
Choose all that is true.

Q22 What is valid C# code below?
   a) `(int x) => x%2==0`
   b) `x => x%2==0`
   c) `var f = x => x*2 + 43.0`
Choose all that is true.
C# Quiz – Type Inference

Q23 What is the type of the anonymous method:
   a) `Action<int>`
   b) `Func<int, int>`  
   c) `Action<int, int>`
   
   \[
   x \Rightarrow x \ast 2 + 43
   \]

Q24 What is the type of the anonymous method:
   a) `Action<int>`
   b) `Func<int, int>`  
   c) `Action<int, int>`
   
   \[
   x \Rightarrow \text{Console.Write}(x \ast 2)
   \]

C# Quiz – Extensions

Q25 What is true about Extensions?
   a) Extensions to classes are defined in `non static` classes
   b) Extensions to classes are defined in `static` classes
   c) Extensions to `non static` classes must be defined `non static`.
   d) Extensions to `static` classes must be defined `static`.
   e) Extensions can be overloaded
   Choose all that is true.

Q26 How can you see that the code below is an extension method and to what type and how many arguments does the method `Print` take?

```csharp
static class BoolExtensions {
    public static String Print(this bool b) {
        return(b ? "ja" : "nej");
    }
}
```
C# Quiz – Extensions

Q27 Does Extensions work on interfaces?
   a) Yes
   b) No

The answer is crucial for LINQ

C# Quiz – Anonymous Objects

Q28 Look at the code below. What is true?
   a) It will compile?
   b) If it will not compile, then why?
   c) Can the type of p1 be expressed in C#?
   d) Can p1 and p2 be compared?
   e) Does the order of fields matter when comparing p1 and p2?

Choose all that is true – if b) is true then assume the malicious statement is removed from the program.

```csharp
var p1 = new { x = 1, y = "foo", z = 3.14 };
p1.x = 42;
var p2 = new { x = 42, y = "foo", z = 3.14 };
Console.WriteLine("p1 = p2 : " + (p1 == p2));
```
C# Quiz – Just Wierd

Q29 Look at the code below.

a) Will it compile?
b) Will it run?

```csharp
public interface I1 {}
public class Foo1 : I1 {}
public class Foo2 {}
class Program {
    static void Main(string[] args) {
        IEnumerable<I1> foo = new Foo1[] { new Foo1(), new Foo1() };
        foreach (Foo2 x in foo) {
            Console.WriteLine(x);
        }
    }
}
```