Revature

# Day 2 .Net

<Spaced out and forgot to start notes>

## .Net

-Cross platform/ Platform specific

>Write code once and implement in different Environments

## .Net SDK vs Runtime

-Software Development Kit (SDK) -**Read and Write**

>Includes everything you need to both build and run application

-Runtime **-Read only**

>Includes just the resources you need to run your application.

>Runtime is also included in the SDK

## Visual Representation of .Net5 (Review recording for imaging)

-.Net5

>Runtime Components

>Compilers

>Languages

-Tools

>VS (Code/Mac/IDE)

>CLI

## .Net Standard

-Set of APIs that are implemented by .Nets Base Library

## Common Language Runtime (CLR)

-Services

>Memory Management

)Automatic garbage collector

* Deletes variables and storages after it is removed from scope

>JIT compilation *(C# isn’t much different than Java)*

) Assist in Cross Platforming (Visual explanation in Recording, recreation provided in folder)

) IL = Interpreter Language (your .exe files)

) Machine Code = Standard binaries ( 1’s and 0’s)

) C# uses CLR , Java uses JVM

>Exception Handling

) Handles Syntax and Logical errors at compile and runtime

### Another representation of the compilation process as described

.cs file =compiles > .Net Compiler = translate > Intermediate Language = passes > CLR = passes to >

JIT Compiler = compiles > native code (Machine Code I.E Binary)

## Managed Code

-Any code managed by the CLR

>C# is an example

## Unmanaged Code

-Code that is developed outside .Net

>C is an example

-Can still be executed, Wrappers are required

## Garbage Collector

-Maintains your Heap memory

>Checks for objects in the managed heap that are no longer in use, and frees this memory for use.

>The Collector creates generations to track objects in the heap on how often that object is used or is referenced. The longer the generation the more it is referenced.

-Memory Outside of the CLR

>These resources must be handled manually using IDisposable Interface dispose() method

)Database connections is an example of something that must be handled with dispose()

)Classes that require IDisposable, should already inherit this Interface in their own code.

/// End of PP

## Discussions

1. What is an SDK?

SDK handles both the build and runtime of an application.

1. What’s the role of the CLR?

Memory Management IE Garbage collector

Runtime environment offered by .Net, compiles the IL to Machine Code using JIT.

Exception Handling during runtime

Allows for easily integrating between other languages

Helps with scalable and Multithreaded applications

1. What’s the role of the BCL?

Contains common runtime libraries used during C# Development.

AKA Base class library.

1. What is managed code?

Code that is covered under the .NET manager (CLR)

Managed code is anything compliant with the .NET CLR

1. What is unmanaged code?

Code that is not covered or compliant with the .NET CLR

1. What’s the garbage collection process?

Way to make memory management easier. Collect and end processes that are no longer in use

1. **Is managed code and unmanaged code, vocabulary specific to the CLR? (Question is not required knowledge, but knowledge is always good =D! )**

No, … and yes. While the specific verbiage is specific to CLR, the concept is used across other developments like the JVM. Ask other developers, and see what they call it. (Scala, Kotlin are other examples but they must be translated first)