Don Petersen

S5 Discussion Questions

1) What is "polymorphism?" How have you used polymorphism in programs you have written?

Polymorphism is one of the pillars of object oriented programming. It is when objects of derived classes are treated as objects of a base class in method parameters and collections or arrays. (msdn 2014) Polymorphism is used in c# to allow derived classes to override virtual methods that are defined and implemented in a base class. Polymorphism is achieved through inheritance where the type of object determines which version of the method to execute. I have used polymorphism to overridden simple methods so that they can do something different than had been previously done but it isn’t something that I have done on my own. My understanding of polymorphism is mainly conceptual.

2) How are inheritance and overriding related to polymorphism? Can you have one without the other? Why or why not?

Polymorphism is considered the third pillar in object oriented programming after encapsulation and inheritance. This is because polymorphism involves inherited methods from base classes being overridden and used in a derived class. Polymorphism cannot exist without inheritance and overriding. The essence of polymorphism is that “a reference variable can refer to any object created from any class related to it by inheritance.” (Lewis 2007)

3) What are reference variables? How are they used? What do you think is the biggest advantage of reference variables over "normal" variables?

Reference variables are similar to pointer in C++. A reference variable is used to refer any object created from any class related to it by inheritance. Instead of directly containing data, reference variables store references to their data. This is used to allow more than one variable to reference the same object and makes it so operations on one variable do not affect the other. Since a reference variable does not contain its data directly, when you pass a reference variable by value, you can change the data pointed to it by the reference without changing the value of the reference itself.(msdn 2014) This makes reference variables more flexible than “normal variables,” by allowing it to refer to different objects depending on the object passed.

4) In what ways can a thrown exception by handled? What happens if an exception is not caught? Have you used exceptions in the past? Give examples.

Thrown exceptions are handled by the use of try/catch blocks. If an exception is not caught is can cause a run-time error when the program is compiled or can cause your program to crash. I have used exception many times since I started learning to program a year ago. My most recent use of exception handling was in Seminar 3’s programming assignment. I used a try/catch block to handle if the user attempted to enter a letter into the hi-lo game instead of a number. The try block would check the input and if a letter was entered then the catch block would output a warning to the user to not enter letters and then would give the user another chance to guess.

5) What is a "stream?" Give examples of at least two kinds of streams.

“A stream is an ordered sequence of bytes.” (Lewis 2007) The term stream refers to the flow of data from source to destination like water flowing down a stream. Data that is received from the user or location to the program is known as an Input stream, and data that flows from the program to a display or location is called an Output stream. An example of an Input stream is Console.Readline(); this is where the program receives data from the user. An example of an Output stream is Console.Out.Writeline(“hello world”); this stream displays the text “hello world” on the screen.

Microsoft Developer Network (msdn) (2014). Polymorphism(C# Programming Guide) Retrieved from <http://msdn.microsoft.com/en-us/library/ms173152.aspx>

Microsoft Developer Network (msdn) (2014). Passing Reference-Type Parameters(C# Programming Guide). Retrieved from <http://msdn.microsoft.com/en-us/library/s6938f28.aspx>

Lewis, J. (2007). C# Software Solutions: Foundations of Program Design.Boston, MA:Pearson Education.