Programming Fundamentals II Sec. 600

Assignment #3

Due date: 6/27/22 at 11:59 pm

1. (40 points) Briefly define (one to two sentences) each of the following ten terms.

1. Abstract class

*Classes that are to abstracts to construct objects. Can have constructor that is protected(# in UML class diagram)*

1. Abstract method

*Method header with no implementation (no method body).*

1. Number

*A class that is parent of numeric wrapper classes, BigInteger, and BigDecimal, includes methods to convert a value to another type*

1. Interface //is-kind-of relationship

*Further level of abstraction consisting only of abstract methods and provide a contract that all subclasses must adhere to. Multiple inheritance*

1. Default method

*Allow to provide added behaviors interfaces. Used very rarely in program.*

1. Comparable

*Comparable interface that defines compareTo method for comparing objects*

1. Cloneable

*Interface to specify that an object can be cloned. Classes that implement Cloneable are marked as cloneable*

1. Marker interface

*Interfaces with empty bodies are known as marker interfaces*

1. Single inheritance

*Abstract classes/ classes sometimes already have methods implemented inside of them. Allows for only one class to inherit from at one time*

1. Multiple inheritance

*Interface only has abstract methods inside of it with implementations unknown. Since they have no implementation and no risk of colliding implementations, this allows for multiple inheritance*

2. (15 points) What is the benefit of declaring a reference variable using an abstract class in relation to method matching and method binding?

*If the super class doesn’t contain a method that is found in a subclass even when using an Actual type that has the method, the method matching will fail if using a reference variable of the super class. Abstract class can be used to write out an abstract method which will allow for the method matching and binding to pass.*

3. (15 points) Why must methods be public in both the parent class and child class to allow method overriding?

*They must have the same signature type. Public can be overridden. Private cannot be overridden. Static methods are not overridden but are hidden.*

4. (15 points) Why are interfaces generally desired over abstract classes?

*Interfaces are generally desired because they are more flexible and allow for multiple inheritance and abstract classes do not. Interfaces also don’t contain any implementation details. It allows for a lot of freedom to decide how the methods will work.*

5. (15 points) How do abstract classes and interfaces improve the class-design guideline of clarity?

*Abstract classes allow for more concise code and comprehensive code because of the ability to declare reference variables when required.*

*Interfaces offer a further level of abstraction. They also provide a contract that all subclasses must adhere to. They also provide for a more concise and comprehensive code because of multiple inheritance. This allows for classes to be flexible in usage and coding environments.*