Content Objective: Students will begin to work with abstract classes and methods while extending interfaces in Java.

|  |  |
| --- | --- |
| **On the Tech Horizon (10pts.)**  **link to a tech/coding related article or journal no more than one month old (no blogs or reddit clones see below)** | |
| URL: |  |
| Reaction/Commentary: |  |

|  |  |
| --- | --- |
| **Tech Terms and History (20pts.)**  **vocabulary from BJ p.463-512 and The Information Chapter 8 (definition/commentary/significance in your words)** | |
| Syntax to create interface | public interface InterfaceName{  //name of method  } |
| Syntax to implement interface | public class OtherClass implements InterfaceName{  public double nameOfMethod(){  return balance;  }  } |
| Why can an interface NOT be instantiated? | Interfaces are not classes, they don’t have objects that can be instantiated |
| Syntax for constants in Java | All variables in an interface have the public static final modifiers automatically  int final constant = 5;  Constants should be all caps with underscores for spaces |
| Syntax to convert class to interface (2 ways) | Declare an object of the class with the type of the interface (Measurable max = new Measurable[3] when max is a Person) |
| Syntax to cast interface to class | BJ p.472 |
| Syntax for clone Method | BJ p.475 |
| Syntax for cloneable Interface | BJ p.475 |
| What is a Callback | BJ p.477-478 |
| What is an “inner class” and when/how would you use it? | BJ p.481-482 |
| What is an “anonymous class” and when/how would you use it? | BJ p.482 |
| What is a “mock object” and when/how would you use it? | BJ p.483 |
| What is an event listener? | BJ p.485 |
| Syntax for creating an inner class listener | BJ p.487 |
| Syntax for mousePress() | BJ p.498 |
| Summarize Turing’s downfall in 50-100 words. | TI p.254-256 |
| Summarize Shannon’s advancements through the 50’s and 60’s in 50-100 words. | TI p.259-268 |

|  |  |
| --- | --- |
| **Code Snippets (30pts.)**  **only submit snippets or classes no full programs required (test and run in IDE, then copy/paste applicable code frag)** | |
| E10.3-10.4 | E10.3  public class Person implements Measurable{  private String name;  private double height;    public Person(String name, double height){  this.name = name;  this.height = height;  }    public double getHeight(){  return height;  }    public String getName(){  return name;  }      //Must have getMeasure to compile if you're implementing Measurable  public double getMeasure(){  return height;  }  }  E10.4  public static Measurable max(Measurable[] objects){  if(objects.length==0){  return null;  }  Measurable max = objects[0];  for(Measurable obj: objects){  if(obj.getMeasure() > max.getMeasure()){  max = obj;  }  }  return max;  } |
| E10.9 |  |
| E10.10-E10.11 |  |
| E10.13-E10.17 |  |
| E10.21-E10.22 |  |

|  |  |
| --- | --- |
| **Code Challenge (30pts.)**  **full functioning application sent to** [**GitHub**](https://github.com/SkylineHigh/CSAdvanced/tree/master/06%20Abstract%20Data%20Types) | |
| You may choose any one of the following code challenges: P10.8-P10.21 and post code to GitHub under “Abstract Data Types” | |
| Notes: |  |

|  |  |
| --- | --- |
| **Badge Progress (10pts.)**  **building your coding profile: Java coding training site to earn badges (recommended site** [**http://coderbyte.com**](http://coderbyte.com) **)** | |
| Screenshot/URL: |  |
| Notes/Issues: |  |

|  |  |
| --- | --- |
| **Notes**  **your notes** | |
| Notes: |  |