Java classes

## Exercise 11.1

## **static** **private** **int** studentCount = 0;

1. Instance variable- belongs to an object and allocated storage with it is created. A static variable -is in memory every time the class is running.
2. The static show come after the public
3. So the constants can be called in any class like :System.out.print(pizza.large()); prints 8.

## Exercise 11.3

1. An interface is part of software that interacts with human users. An Interface provides information needed to use a class without revealing the complained stuff or making the user call anything. Has no code just classes methods, could also be called a blueprint.

## Exercise 11.4

**public** **interface** Account {

**public** **void** deposit(**double** dep);

**public** **void** withdrawal(**double** withdraw);

**public** **double** getBalance();

}

1. All the method defined in the interface

## Exercise 11.5

1. Classes inherits it characteristics from the classes above in the hierarchy.
2. Expand <interface>
3. Super activates the constructor and the constructor determines the parameter list
4. Have to call super dot method name.
5. Can’t be change and use only in low hierarchy then the current class.
6. Shape s = new Wheel();

(wheel)s.setSpokes(5);

## Exercise 11.6

1. A. No error

B. Should not cast Type string

C. Nothing is named string

D. nothing is named string

## Exercise 11.7

1. A class that cannot be instantiated and contains variables and methods in common from it super classes.
2. Some methods are different in the subclasses and share no code in common, also cannot be written any code to, so abstract is included in their headers.
3. A final method is a method that is declared as such to It will not be overwritten in its subclasses

Example:

Public final double getXPos(){

Return xPos;

}

1. The variable can only be use in the class it is currently in or its subclasses.

Example:

Protected double xPos;

## Exercise 11.9

1. When an object of a is expected, its always acceptable to substitute an object of a subclass but never its superclasses. Subclasses inherits all the methods of its parent class.

2 B is the only correct because M can’t be sent anything higher then BB

## Exercise 11.10

1. Describes what is true before a method is called. Preconditions describes the value of parameters and instance variables about to be used.

If (I < 1|| I > tests.length ||score < 0 || score >100)

1. Describes what will be true after the program is done executing. The post conditions escribe the return value and changes made to instance variables.

Test[i-1] =score;

## Exercise 11.11

1. Java throws an exception to signal a run-time error. Java will throw an arithmetic exception when a program attempts to divide by 0.
2. If no try-catch method is located, the computer halts the program with a trace of the method calls, the type of exception, and its error message.
3. Using a if-else statement to ask the right question before a method is called, or using a try-catch statement to catch any exceptions that the method might throw and respond to them gracefully.
4. @param used to denote a description of the parameter a method can receive. @return used to describe return values. @throws shows that there’s a check for an exception.

## Exercise 11.12

1. == compares the references to the object in memory, while “equals” check to see if the object don’t have the same location in memory, but the characters are the same.
2. false

True

True

1. Assigned that object to same place in memory.
2. Make a copy method like this:

Public Object clone(){

Return new Student(name,tests);

}