Submit answers for all of the following questions.

EXERCISES:

Chapter 14: 3, 5, 6, 8

3. Starts with sparky which is an object in the dog class. Then it sees println() where “+sparky” is present which runs a toString() which returns bark, bark. So it says “sparky = bark, bark”. Then it tries to print the next part which is “lassie =” unfortunately this will then bring up the objects version of toString so it prints “lassie = Animal@1db9742”.

5. The binding takes place at the last possible moment, right before the method is executed so it is called late binding.

6.

Animal[] animal= New Animal[2];

animal[0] = new Dog();

animal[1] = new Dog();

**Dog[] fido = animal[0]**

**Dog[] sparky = animal[1];**

Compile time error

8.

Animal[] animals = new Animal[20];

animals[0] = new Dog();

animals[1] = new Cat();

Dog lassie = (Dog) animals;

Cat fluffy = (Cat) animals;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* InsuranceDriver.Java

\* <Adam Cox>

\*

\* This is the driver for Insurance

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** java.util.\*;

**public** **class** InsuranceDriver

{

**public** **static** **void** main(String[] args)

{

InsuranceLife lifePolicy; // holds a life insurance policy

InsuranceHealth healthPolicy; // holds a health insurance policy

**double** actualAdjustment; // actual amount adjusted for premium

// Array of insurance policies, to be filled with both types

InsurancePolicy[] policies = **new** InsurancePolicy[50];

lifePolicy = **new** InsuranceLife("John Dean", 40, 400000);

lifePolicy.addBeneficiary("Caiden Dean");

lifePolicy.addBeneficiary("Jordan Dean");

lifePolicy.adjustCoverage(+10);

policies[0] = lifePolicy;

healthPolicy = **new** InsuranceHealth("John Dean", 500, 2000);

healthPolicy.adjustCoverage(+100);

policies[1] = healthPolicy;

healthPolicy = **new** InsuranceHealth("Taylor Swift", 600, 1000);

actualAdjustment = healthPolicy.adjustCoverage(-700);

System.***out***.println("Actual premium adjustment = " + actualAdjustment);

policies[2] = healthPolicy;

**for** (**int** i=0; i<policies.length && policies[i] != **null**; i++)

{

policies[i].display();

System.***out***.println();

}

} // end main

} // end of InsuranceDriver

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* InsurancePolicy.java

\* <Adam Cox>

\*

\* This is for the class InsurancePolicy.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **abstract** **class** InsurancePolicy

{

**protected** String policyHolder; // Policy Holder's Name

**protected** **double** premium = 0; // Premium for adjustment

**public** **abstract** **void** display();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** InsurancePolicy(String name, **double** premiumValue)

{

**this**.policyHolder = name;

**this**.premium = premiumValue;

} // end of constructor

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// This determines premium changes.

**public** **double** adjustCoverage(**double** premiumChange)

{

**if**((premium+premiumChange)>10) // If premium changes are above minimum

{

premium+=premiumChange;

**return** premiumChange;

}

**else** // else premium changes are below minimum

{

**double** oldPremium=premium; // Stores the old Premium for adjustment

premium=10;

**return** 10-oldPremium;

}

} // end of adjustCoverage

} // end of InsurancePolicy

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* InsurancePolicy.java

\* <Adam Cox>

\*

\* This is for the class InsurancePolicy.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **abstract** **class** InsurancePolicy

{

**protected** String policyHolder; // Policy Holder's Name

**protected** **double** premium = 0; // Premium for adjustment

**public** **abstract** **void** display();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** InsurancePolicy(String name, **double** premiumValue)

{

**this**.policyHolder = name;

**this**.premium = premiumValue;

} // end of constructor

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// This determines premium changes.

**public** **double** adjustCoverage(**double** premiumChange)

{

**if**((premium+premiumChange)>10) // If premium changes are above minimum

{

premium+=premiumChange;

**return** premiumChange;

}

**else** // else premium changes are below minimum

{

**double** oldPremium=premium; // Stores the old Premium for adjustment

premium=10;

**return** 10-oldPremium;

}

} // end of adjustCoverage

} // end of InsurancePolicy

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* InsuranceHealth.java

\* <Adam Cox>

\*

\* This is for InsuranceHealth, subclass of InsurancePolicy.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** InsuranceHealth **extends** InsurancePolicy

{

**private** **double** deductible; //Record the deductible.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** InsuranceHealth(String name, **double** premiumValue, **double** deductibleValue)

{

**super**(name, premiumValue);

**this**.deductible=deductibleValue;

} // end of constructor

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// This determines the changes to premium and deductible

**public** **double** adjustCoverage(**double** deductibleChange)

{

**double** adjustmentValue = 0; //used for adjustment recording

adjustmentValue=**super**.adjustCoverage(deductibleChange);

deductible -= adjustmentValue\*10;

**return** adjustmentValue;

} // end of adjustCoverage

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// This displays the Insured's information

**public** **void** display()

{

System.***out***.println("Health insurance policy holder: "+**super**.policyHolder);

System.***out***.printf("Premium: $%,.2f\n"

+ "Deductible: $%,.2f\n",**super**.premium,**this**.deductible);

} // end of display

} // end of InsuranceHealth

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* InsuranceLife.java

\* <Adam Cox>

\*

\* This is for InsuranceLife, subclass of InsurancePolicy.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** java.util.\*;

**public** **class** InsuranceLife **extends** InsurancePolicy

{

**private** **double** deathBenefit; // For Death Benefits

ArrayList<String> beneficiaries = **new** ArrayList<String>(); // People that get money

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**public** InsuranceLife(String name, **double** premiumValue, **double** deathPayday)

{

**super**(name, premiumValue);

**this**.deathBenefit = deathPayday;

} // end of constructor

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// This determines the changes to premium and benefit

**public** **double** adjustCoverage(**double** premiumChange)

{

**double** adjustmentValue = 0;

adjustmentValue = **super**.adjustCoverage(premiumChange);

deathBenefit += premiumChange\*10000;

**return** adjustmentValue;

} // end of adjustCoverage

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// This displays the Insured's information

**public** **void** display()

{

System.***out***.println("Life insurance policy holder: "+**super**.policyHolder);

System.***out***.printf("Premium: $%,.2f\n"

+ "Death benefit: $%,.2f\n",**super**.premium,**this**.deathBenefit);

System.***out***.println("Beneficiaries:");

**for** (String receiver : beneficiaries)

{

System.***out***.println(receiver);

} // end of display

} // end of InsuranceLife

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// This adds a beneficiary

**public** **void** addBeneficiary(String moneyReceiver)

{

beneficiaries.add(moneyReceiver);

} // end of addBeneficiary

} // end of InsuranceLife

Actual premium adjustment = -590.0

Life insurance policy holder: John Dean

Premium: $50.00

Death benefit: $500,000.00

Beneficiaries:

Caiden Dean

Jordan Dean

Health insurance policy holder: John Dean

Premium: $600.00

Deductible: $1,000.00

Health insurance policy holder: Taylor Swift

Premium: $10.00

Deductible: $6,900.00