**COMP 132 - Homework**

**Inheritance and Polymorphism**

1. This question builds upon the TextMessage example from class contained in the inheritance package in the provided code. You will define a new class named PaymentMessage that represents a new type of text message that can be sent to a vendor to make a payment. This new type of message has text but also has a cash value.

a. Give an implementation of the PaymentMessage class. Because a PaymentMessage is everything a TextMessage is, plus some other stuff, it should be a subclass of TextMessage. In addition to being a subclass of TextMessage, the PaymentMessage must:

i. have a field to keep track of the amount of the payment

ii. have an appropriate constructor

iii. have an accessor for the amount of the payment.

iv. not implement any other methods (yet).

b. List all of the methods that can be invoked on a PaymentMessage object as defined in part a. Remember a subclass inherits methods from its superclass (which also inherits from its superclass!).

c. Consider the following snippet of code that uses a PaymentMessage object as defined in part a:

PaymentMessage pm = **new** PaymentMessage(

7173456789L, 71798765432L, "Here ya go", 22.75);

System.*out*.println("pm.getMessageType(): " +

pm.getMessageType());

When executed this code would produce the output:

pm.getMessageType(): Text Message

Which is clearly not the desired output. Explain as clearly and fully as you can why this snippet generates this output.

d. Give the code that you would add to the PaymentMessage class to cause the snippet of code in part c produce the output:

pm.getMessageType(): Payment Message

e. Currently the length of a PaymentMessage would simply be the length of the text that it contains. However, storing the payment requires some space as well. Let us assume that the size of a PaymentMessage should be the length of its text plus 8 units of storage for the payment information. (The usual encoding of a double-precision floating point number requires eight bytes, but we do not study those details here.) Give the code that you would add to the PaymentMessage class so that it computes the correct length.

2. The questions below make use of the following three classes:

**class** Ecks {

**private** **int** x;

**public** Ecks(**int** a) {

x = a;

}

**public** **int** bar() {

**return** x + 1;

}

**public** **int** foo() {

**int** b = bar();

**return** x\*b;

}

}

**class** Why **extends** Ecks {

**private** **int** y;

**public** Why(**int** b) {

**super**(7);

y = b;

}

**public** **int** bar() {

**int** c=**super**.bar();

**return** c + y;

}

**public** **int** qux() {

**return** y + 3;

}

}

**class** Zee **extends** Why {

**private** **int** z;

**public** Zee() {

**super**(5);

z = 3;

}

**public** **int** bar() {

**int** d = qux();

**return** z \* d;

}

}

Assume that the following statements are executed before each of the questions below (i.e. the parts below do not build upon each other, each is started with a fresh set of objects and references).

Ecks xx = **new** Ecks(2);

Why yy = **new** Why(3);

Zee zz = **new** Zee();

Ecks xx2 = yy;

Ecks xx3 = zz;

Note: Try to answer the following questions without compiling or running the code.

a. For each of the following statements, indicate if it is *legal* (i.e. will compile as written) or *illegal* (will generate a compiler error):

i. Ecks x1 = yy;

ii. Why y1 = xx;

iii. Why y2 = zz;

iv. Zee z2 = yy;

b. For each of the following statements, indicate if it is *legal* (i.e. will compile as written) or will generate a *compile time error* or a *runtime error*.

i. Why y2 = xx3;

ii. Why y3 = (Why)xx3;

iii. Zee z1 = (Zee)xx3;

iv. Zee z2 = (Zee)xx2;

v. Why y4 = (Zee)xx3;

c. Give the output that would be produced by the following lines of code:

System.*out*.println("xx.bar() = " + xx.bar());

System.*out*.println("yy.bar() = " + yy.bar());

d. Give the output that would be produced by the following lines of code:

System.*out*.println("xx.foo() = " + xx.foo());

System.*out*.println("yy.foo() = " + yy.foo());

e. Give the output that would be produced by the following lines of code:

System.*out*.println("zz.bar() = " + zz.bar());

System.*out*.println("zz.foo() = " + zz.foo());