# Polymorphism

Complete the questions below with the aid of your textbook, notebook, or classroom resources. You must work on the problem on your own. When your instructor calls “time,” pass your paper to the person next to you, and accept the new worksheet. You should first work on the next blank problem on the sheet, then, if you have time, scan through the sheet to see if you spot any errors in others’ work. Correct those errors if you have time, then pass the sheet again the next time your instructor calls “time.”

1. Using the *A*, *B*, *C*, and *D* classes from today’s lesson, and the hierarchy and tables you drew in your notebook, predict the output of the following code fragment:

public static void main(String[] args){

A[] elements = {new B(), new A(), new C()};

for (int i = 0; i < elements.length; i++){

elements[i].method2();

System.out.println(element[i]);

elements[i].method1();

System.out.println();

}

}

1. When should you use the super keyword?
2. Assume that the following classes have been defined:

public class Flute extends Blue{

public void method2(){

System.out.println(“flute 2”);

}

public String toString(){

return “flute”;

}

}

public class Blue extends Moo{

public void method1(){

System.out.println(“blue 1”);

}

}

public class Shoe extends Flute{

public void method1(){

System.out.println(“shoe 1”);

}

}

public class Moo{

public void method1(){

System.out.println(“moo 1”);

}

public void method2(){

System.out.println(“moo 2”);

}

public String toString(){

return “moo”;

}

}

What is the output produced by the following code fragment?

public static void main (String[] args){

Moo[]elements = {new Shoe(), new Flute(), new Moo(), new Blue()};

for (int i = 0; i < elements.length; i ++){

System.out.println(elements[i]);

elements[i].method1();

elements[i].method2();

System.out.println();

}

}

You should use the space below to draw your class hierarchy and method/class output table.

1. Using the classes from the previous problem, write the output that is produced by the following code fragment:

public static void main(String[] args){

Moo[] elements = {new Blue(), new Moo(), new Shoe(), new Flute()};

for (int i = 0; i < elements.length; i++){

elements[i].method2();

elements[i].method1();

System.out.println(elements[i]);

System.out.println();

}

}

1. What is the difference between overloading and overriding a method?
2. Assume that the following classes have been defined:

public class AquaticMammal extends SeaCreature{

public void method1(){

System.out.println(“warm-blooded”);

}

}

public class SeaCreature{

public void method1(){

System.out.println(“creature 1”);

}

public void method2(){

System.out.println(“creature2”);

}

public String toString(){

return “ocean-dwelling”;

}

}

public class Whale extends AquaticMammal {

public void method1(){

System.out.println(“spout”);

}

public String toString(){

return “BIG!”;

}

}

public class Squid extends SeaCreature{

public void method2(){

System.out.println(“tentacles”);

}

public String toString(){

return “squid”;

}

}

What output is produced by the following code fragment?

public static void main(String[] args){

SeaCreature[] elements = {new Squid(), new Whale(), new SeaCreature(), new AquaticMammal());

for (int i = 0; i < elements.length; i++){

System.out.println(elements[i]);

elements[i].method1();

elements[i].method2();

System.out.println();

}

}

You should use the space below to draw your class hierarchy and method/class output table.

1. Using the classes from the previous problem, write the output that is produced by the following code fragment:

public static void main(String[] args){

SeaCreature[] elements = {new SeaCreature(), new Squid(), new AquaticMammal(), new Whale()};

for (int i = 0; i < elements.length; i++){

elements[i].method2();

System.out.println(elements[i]);

elements[i].method1();

System.out.println();

}

}

1. Assume that the following classes have been defined:

public class Bay extends Lake{

public void method1(){

System.out.print(“Bay 1 ”);

super.method2();

}

public void method2(){

System.out.print(“Bay 2 ”);

}

}

public class Pond{

public void method1(){

System.out.print(“Pond 1 “);

}

public void method2(){

System.out.print(“Pond 2 “);

}

public void method3(){

System.out.print(“Pond 3 “);

}

}

public class Ocean extends Bay{

public void method2(){

System.out.print(“Ocean 2 “);

}

}

public class Lake extends Pond {

public void method3(){

System.out.print(“Lake 3 “);

method 2();

}

}

What output is produced by the following code fragment?

Pond[] ponds = {new Ocean(), new Pond(), new Lake(), new Bay()};

for(Pond p : ponds){

p.method1();

System.out.println();

p.method2();

System.out.println();

p.method3();

System.out.println(“\n”);

}

You should use some of the space below to draw your class hierarchy and method/class output table.

1. Suppose that the following variables referring to the classes from the previous problem are declared:

Pond var1 = new Bay ();

Object var2 = new Ocean();

Does the following statement cause a compiler error? If the statement does not produce an error, what is the output of the statement?

((Lake) var1).method1();

1. Refer to the code included in questions 8 and 9 when answering this question. Does the following statement cause a compiler error? If the statement does not produce an error, what is the output of the statement?

((Bay) var1).method1();

1. Refer to the code included in questions 8 and 9 when answering this question. Does the following statement cause a compiler error? If the statement does not produce an error, what is the output of the statement?

((Pond) var2).method2();

1. Refer to the code included in questions 8 and 9 when answering this question. Does the following statement cause a compiler error? If the statement does not produce an error, what is the output of the statement?

((Lake) var2).method2();

1. Refer to the code included in questions 8 and 9 when answering this question. Does the following statement cause a compiler error? If the statement does not produce an error, what is the output of the statement?

((Ocean) var2).method3();