Lab 2

Classes and Objects

1. Inheritance. (5 points, 1 point per question)

Read the review questions at <http://javaconceptoftheday.com/java-inheritance-practice-coding-questions/> . Then, answer the following questions.

Given the following class definitions:

public class Mouse { // more code }

public class USB\_Mouse extends Mouse { // more code }

public class Wireless\_BT\_Mouse extends Mouse { // more code }

* 1. What keyword indicates USB\_Mouse is a type of Mouse?

*-extends*

* 1. Is a Wireless\_BT\_Mouse a type of Mouse or a type of USB\_Mouse?

*-both*

* 1. Is Mouse a superclass or a subclass?

*-superclass*

* 1. What is the superclass of USB\_Mouse?

*-Mouse*

* 1. Suppose the main program instantiates Wireless\_BT\_Mouse like so:  
     Wireless\_BT\_Mouse MBT = new Wireless\_BT\_Mouse();

If all three classes have no-arg constructors, which constructor is executed first: Wireless\_BT\_Mouse() or Mouse()?

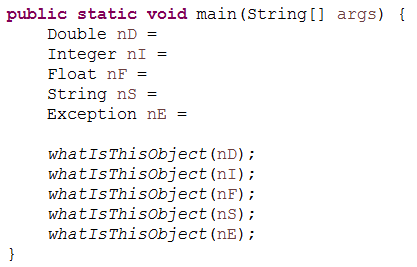
*-Mouse()*

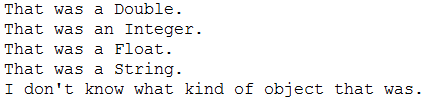
2. Polymorphism / instanceOf. (10 points)

Write the method whatIsThisObject. whatIsThisObject accepts a single Object as a parameter, and outputs to the console based on what type of object is passed. Create a main program that tests whatIsThisObject for all the wrapper classes listed below. If a different kind of object is passed, the method should print “I don’t know what kind of object that was.” to the console.

|  |
| --- |
| **Wrapper Class** |
| Boolean |
| Byte |
| Character |
| Integer |
| Float |
| Double |
| Long |
| Short |

A sample main (test) program and output follow. **This program does not test all the required data types. You must supply your own initial values for test variables. You must test whatIsThisObject with an object of type Exception in addition to the wrapper classes listed here.**





Rubric:  
Student name and today’s date is a comment in the first line of the programs: -5 points if fails  
whatIsThisObject works as described: 4 points  
Instantiate/test eight wrapper class objects plus an Exception object: 4 points  
Program structure: 2 points

Please paste a screenshot of a successful program run, and copy-and-paste the source code from your .java file, here:



**Solution.java:**

public class Solution {  
  
 public static void main(String[] args) {  
 Boolean nBool = true;  
 Character nC = 'a';  
 Byte nB = 100;  
 Short nS = 1000;  
 Integer nI = 10000;  
 Long nL = 1000000000L;  
 Float nF = 1.1f;  
 Double nD = 10.1;  
 String nX = "¯\\\_(ツ)\_//¯";  
  
 *whatIsThisObject*(nBool);  
 *whatIsThisObject*(nC);  
 *whatIsThisObject*(nB);  
 *whatIsThisObject*(nS);  
 *whatIsThisObject*(nI);  
 *whatIsThisObject*(nL);  
 *whatIsThisObject*(nF);  
 *whatIsThisObject*(nD);  
 *whatIsThisObject*(nX);  
 }  
  
 private static void whatIsThisObject(Object obj) {  
 if (obj instanceof Boolean) {  
 System.*out*.println("That was a boolean.");  
 } else if (obj instanceof Character) {  
 System.*out*.println("That was a character.");  
 } else if (obj instanceof Number) {  
 if (obj instanceof Byte) {  
 System.*out*.println("That was a byte.");  
 } else if (obj instanceof Short) {  
 System.*out*.println("That was a short.");  
 } else if (obj instanceof Integer) {  
 System.*out*.println("That was an integer.");  
 } else if (obj instanceof Long) {  
 System.*out*.println("That was a long.");  
 } else if (obj instanceof Float) {  
 System.*out*.println("That was a float.");  
 } else if (obj instanceof Double) {  
 System.*out*.println("That was a double.");  
 }  
 } else {  
 System.*out*.println("I don't know what that object is.");  
 }  
 }  
}

3. Interfaces. Based on Reges self-check 9.5.20. (10 points)

Consider the following interface and class:

public interface I {

public void m1();

public void m2();

}

public class C implements I {

// code for class C

}

1. What must be true about the code for class C in order for that code to compile successfully? (2 points)

*-The code must implement both methods in the interface; m1 and m2.*

2. Create I.java, C,java, and a main program class, to meet the following requirements:

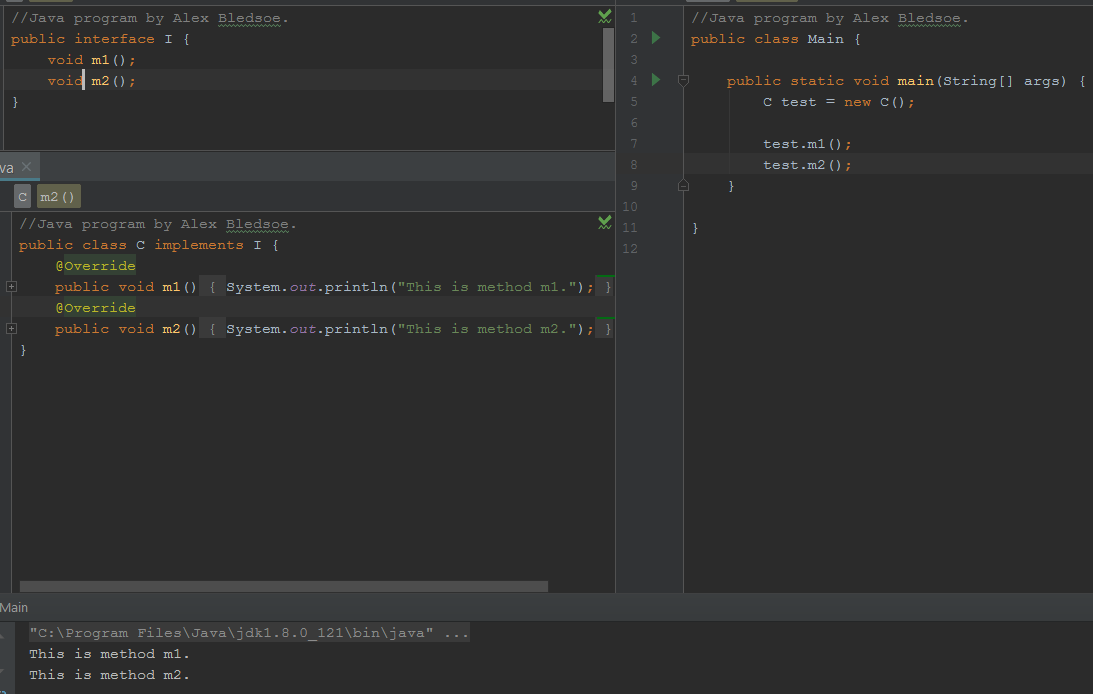
2.1. Your name and today’s date is a comment on the first line of all three files.

2.2. I.java has exactly the code listed in the example.

2.3. C.java implements methods m1() and m2():  
 method m1() outputs “This is method m1” to the console.  
 method m2() outputs “This is method m2” to the console.  
2.4. The main program instantiates class C and calls method m1 and m2 of the instantiated object.

Please paste a screenshot of a successful program run, and copy-and-paste the source code from **all three** .java files, here:

Rubric:  
Student name and date is in a comment on the first line of the programs: -5 points if fails  
Question 1: 2 points  
Three .java files: 3 points  
I.java: 1 point  
C. java: 2 points  
Main java program: 2 points



**Main.java:**

//Java program by Alex Bledsoe.  
public class Main {  
  
 public static void main(String[] args) {  
 C test = new C();  
  
 test.m1();  
 test.m2();  
 }  
  
}

**I.java:**

//Java program by Alex Bledsoe.  
public interface I {  
 void m1();  
  
 void m2();  
}

**C.java:**

//Java program by Alex Bledsoe.  
public class C implements I {  
 @Override  
 public void m1() {  
 System.*out*.println("This is method m1.");  
 }  
  
 @Override  
 public void m2() {  
 System.*out*.println("This is method m2.");  
 }  
}

4. Generics. (5 points, 1 point per question)

Thanks to <https://www.tutorialspoint.com/java/java_generics.htm> .

Create the following Java class to demonstrate Generics. Please remember to type in this code rather than copy-and-pasting; Word loves to mangle source code.

// Student Name Today’s Date

public class MaximumTest {

// determines the largest of three Comparable objects

public static <T extends Comparable<T>> T maximum(T x, T y, T z) {

T max = x; // assume x is initially the largest

if(y.compareTo(max) > 0) {

max = y; // y is the largest so far

}

if(z.compareTo(max) > 0) {

max = z; // z is the largest now

}

return max; // returns the largest object

}

public static void main(String args[]) {

System.out.printf("Max of %d, %d and %d is %d\n\n",

3, 4, 5, maximum( 3, 4, 5 ));

System.out.printf("Max of %.1f,%.1f and %.1f is %.1f\n\n",

6.6, 8.8, 7.7, maximum( 6.6, 8.8, 7.7 ));

System.out.printf("Max of %s, %s and %s is %s\n","pear",

"apple", "orange", maximum("pear", "apple", "orange"));

}

}

4.1. What data type is T when maximum is called with 3, 4, 5?

*-Integer*

4.2. What data type is T when maximum is called with 6.6, 8.8, 7.7?

*-Double*

4.3. What data type is the variable max when maximum is called with “pear”, “apple”, “orange”?

*-String*

4.4. Suppose maximum was defined as:

public static <E extends Comparable<E>> E maximum(E x, E y, E z)

Which definition for max is correct?

a. E max = x;

b. T max <E> = x;

c. x max = new E();

d. T max = new E();

4.5. Why does maximum use compareTo instead of ==? (Hint: why can’t you use == with object reference variables?

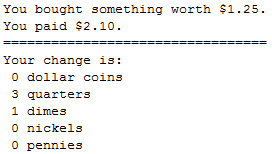
*-“==” compares the object references, not the objects themselves.*

5. Loop, input, and variable review: objects not required. Based on Goodrich 6th edition, P-2.36. (15 points)

Write a Java program that can make change for a vending machine. Your program should take two numbers as input, one for the amount charged and one for the money given. You can assume both amounts are less than $10, and that the user will give more money than the amount due.

Use the following U.S. coins: dollar, quarter, dime, nickel, penny. The program displays the number of each kind of coin to give back as. Try to design your program so that it returns the fewest number of coins possible.

Output of a sample program run:

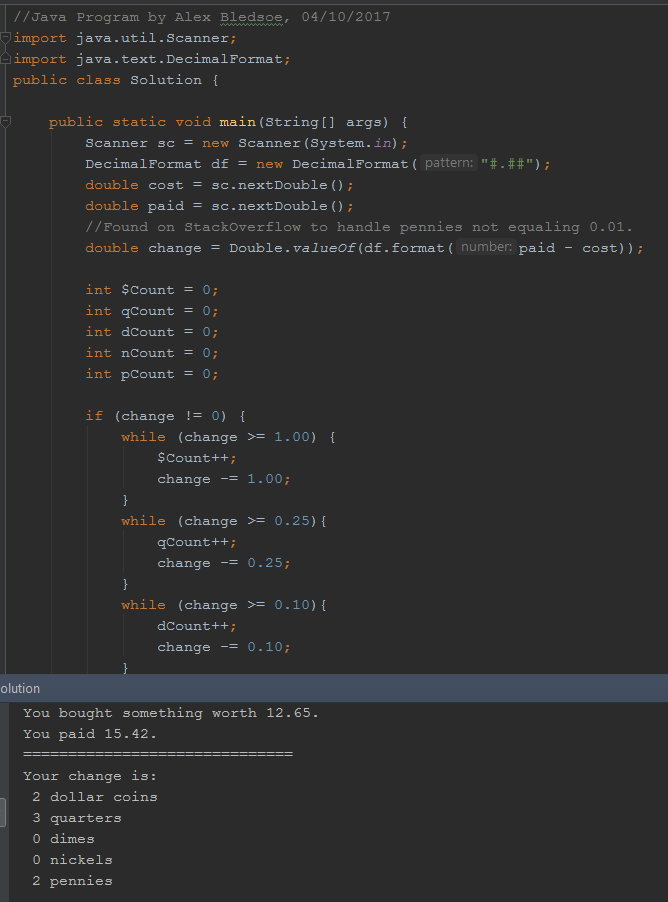


Rubric:

Student name and today’s date are a comment on the first line of the program (-5 if fails)

Input two numbers: 2 points  
Change due calculation: 1 point  
Correct change: 5 points  
Minimum coins correct change: 5 points  
Overall program operation: 2 points

Please paste a screenshot of a successful program run, and copy-and-paste the source code from both .java files, here:



**Solution.java:**

//Java Program by Alex Bledsoe, 04/10/2017  
  
import java.util.Scanner;  
import java.text.DecimalFormat;  
  
public class Solution {  
  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 DecimalFormat df = new DecimalFormat("#.##");  
 double cost = sc.nextDouble();  
 double paid = sc.nextDouble();  
 //Found on StackOverflow to handle pennies not equaling 0.01.  
 double change = Double.*valueOf*(df.format(paid - cost));  
  
 int $Count = 0;  
 int qCount = 0;  
 int dCount = 0;  
 int nCount = 0;  
 int pCount = 0;  
  
 if (change != 0) {  
 while (change >= 1.00) {  
 $Count++;  
 change -= 1.00;  
 }  
 while (change >= 0.25) {  
 qCount++;  
 change -= 0.25;  
 }  
 while (change >= 0.10) {  
 dCount++;  
 change -= 0.10;  
 }  
 while (change >= 0.05) {  
 nCount++;  
 change -= 0.05;  
 }  
 while (change >= 0.01) {  
 pCount++;  
 change -= 0.01;  
 }  
 }  
 System.*out*.printf("You bought something worth %.2f.\r\n", cost);  
 System.*out*.printf("You paid %.2f.\r\n", paid);  
 System.*out*.println("==============================");  
 System.*out*.println("Your change is:");  
 System.*out*.println(" " + $Count + " dollar coins");  
 System.*out*.println(" " + qCount + " quarters");  
 System.*out*.println(" " + dCount + " dimes");  
 System.*out*.println(" " + nCount + " nickels");  
 System.*out*.println(" " + pCount + " pennies");  
 }  
}