

Home Assignment 6

CPE 102

Chapter 10

1. Suppose you want to use the average method to find the average salary of an array of Employee objects. What condition must the Employee class fulfill?

The employee class has to be able use the measurable interface and measure on the salary. Then we can use the existing methods within this to get the average salary.

2. Why can't the average method have a parameter variable of type Object[]?

The object class would need to have the getAverage method.

3. Why can't you use the average method to find the average length of String objects?

Because you would need to implement in measurable interface on string objects differently than other measurable things.

4. What is wrong with this code?

```
Measurable meas = new Measurable();  
System.out.println(meas.getMeasure());
```

Measurable isn't a class, it's an interface so you can't have something of that type.

5. What is wrong with this code?

```
Measurable meas = new Country("Uruguay", 176220);  
System.out.println(meas.getName());
```

There is no getName method in measurable.

6. Can you use a cast (BankAccount) meas to convert a Measurable variable meas to a BankAccount reference?

Only if it is a valid cast, meaning the meas is a BankAccount object.

7. If both BankAccount and Country implement the Measurable interface, can a Country reference be converted to a BankAccount reference?

No, Country may implement the measurable interface but they are still different objects.

8. Why is it impossible to construct a Measurable object?

Because you can't have an interface object.

9. Why can you nevertheless declare a variable whose type is Measurable?

Because the variable is a reference to an object of a class that implements Measureable.

10. What does this code fragment print? Why is this an example of polymorphism?

```
Measurable[] data = { new BankAccount(10000), new Country("Belgium", 30510) };  
System.out.println(average(data));
```

It prints 20255, the average of 10000 and 30510. While "valid" code, it's not actually meaningful as the two getMeasurment calls are different things.

11. How can you sort an array of Country objects by increasing area?

By implementing the comparable interface then using the existing array.sort methods.

12. Can you use the Arrays.sort method to sort an array of String objects? Check the API documentation for the String class.

Yes, string implements comparable so it can be used.

13. Can you use the Arrays.sort method to sort an array of Rectangle objects? Check the API documentation for the Rectangle class.

No, comparable is not implemented for rectangles.

14. Write a method max that finds the larger of any two Comparable objects.

```
public static Comparable max( Comparable x, Comparable y) {  
    if (x.compareTo(y) > 0) { return x; }  
    else { return y; }  
}
```

15. Write a call to the method of Self Check 14 that computes the larger of two bank accounts, then prints its balance.

```
BankAccount biggest = (BankAccount) max(x, y);  
System.out.println(biggest.getBalance());
```

Chapter 11

1. What happens when you supply the same name for the input and output files to the Total program? Try it out if you are not sure.

You lose the input file as soon as the first stream is opened to the output file.

2. What happens when you supply the name of a nonexistent input file to the Total program? Try it out if you are not sure.

The program would throw an error.

3. Suppose you wanted to add the total to an existing file instead of writing a new file. Self Check 1 indicates that you cannot simply do this by specifying the same file for input and output. How can you achieve this task? Provide the pseudocode for the solution.

You would need to store all the values from the input file and rewrite them to the final output file.

4. How do you modify the program so that it shows the average, not the total, of the inputs?

With each new value add it to a sum like before but also count the number of values. At the return, divide the sum by the number of entries.

5. How can you modify the Total program so that it writes the values in two columns, like this:

```
32.00 54.00
67.50 29.00
35.00 80.00
115.00 44.50
100.00 65.00
Total: 622.00
```

This can be done by only using println every other print which would be simple to implement using a if loop and a count variable to switch between the two.

6. Suppose the input contains the characters Hello, World!. What are the values of word and input after this code fragment?

```
String word = in.next();
String input = in.nextLine();
Word = Hello,

Input = World!
```

7. Suppose the input contains the characters 995.0 Fred. What are the values of number and input after this code fragment?

```
int number = 0;  
if (in.hasNextInt()) { number = in.nextInt(); }  
String input = in.next();  
Input = 995.0 Fred and number is still 0.
```

8. Suppose the input contains the characters 6E6 6,995.00. What are the values of x1 and x2 after this code fragment?

```
double x1 = in.nextDouble();  
double x2 = in.nextDouble();  
x1 = 6000000 and x2 doesn't work because of the ,
```

9. Your input file contains a sequence of numbers, but sometimes a value is not available and is marked as N/A. How can you read the numbers and skip over the markers?

Read the whole file in as strings and only convert when the string is not N/A

10. How can you remove spaces from the country name in Section 11.2.4 without using the trim method?

By using the substring method and searching for the last whitespace character. After that's found, just trim the extra part of the name off the end.

11. If the program is invoked with `java CaesarCipher -d file1.txt`, what are the elements of args?

Args[0] is `-d` and args[1] is `file1.txt`

12. Trace the program when it is invoked as in Self Check 11.

13. Will the program run correctly if the program is invoked with `java CaesarCipher file1.txt file2.txt -d`? If so, why? If not, why not?

Yes it will. The options can be given in any order.

14. Encrypt CAESAR using the Caesar cipher.

FDHVDU

15. How can you modify the program so that the user can specify an encryption key other than 3 with a `-k` option, for example

```
java CaesarCipher -k15 input.txt output.txt
```

```
else if ( option == 'k'){
```

```
    key = Integer.parseInt(args[1].substring(2));
```

```
Return loves
```