Unit 2: Parameters and Returns

Parameters and Returns

Some methods require a parameter or multiple parameters to run​. When the method is called, the value is stored into the parameter variable and the method's code executes using that value.​

If a method accepts a parameter, it is illegal to call it without passing any value for that parameter.​ **Overloading** is when more than one method in a single class has the same name but different parameters.​ The main purpose of overloading is for code to be easier to read or use.​

Parameters can be changed while in a method, but those changes only live within that method. If you want to know the parameter value if it was changed in the method, then you must **return** a value.

Parameter Syntax

Writing a method with one parameter and calling a method with one parameter:

public static void method1 ( type name ) {​

    statement(s);​

}​

method1 (expression);​

Writing a method with multiple parameters and calling a method with multiple parameters:

public static void method2 ( type1 name1, type2 name2, … ,typeN nameN) {​

    statement(s);​

}​

method2 (expression1, expression2, …, expressionN);​

Common Errors

1. If a method accepts a parameter, it is illegal to call it without passing any value for that parameter.​
2. The value passed to a method must be of the correct type.​
3. When primitive variables (int, double) are passed as parameters, their values are copied. Modifying the parameter will not affect the variable passed in.​

Examples

|  |  |
| --- | --- |
| public static void main(String[] args) {​      greeting(“hello”);​      greeting(“good day!”);​  }​  public static void greeting(String greet){  System.out.println(“The greeting is ”+ greet);  } | The greeting is hello​  The greeting is good day!​ |
| public static void main(String[] args) {​      nameAndAge(“Polly”, 3);​      nameAndAge(“Roger”, 456);​  }​  public static void nameAndAge(String name, int age){  System.out.println(“My name is ”+ name+” and my age is ”+age);  } | My name is Polly and my age is 3  My name is Roger and my age is 456 |
| public static void main(String[] args) {​      greeting();​​  }​  public static void greeting(String greet){  System.out.println(“The greeting is ”+ greet);  } | This is an error |
| public static void main(String[] args) {​     temperature (“Fifty-five degrees”);​     temperature (“55”);​  }​  public static void temperature(String temp){  System.out.println(“The temperature is ”+ temp);  }  public static void temperature(int temp){  System.out.println(“The temperature is ”+ temp);  } | The temperature is Fifty-five degrees  The temperature is 55 |

​​

Math Class

The Math Class is useful for common mathematical calculations. Note that the methods do not print to the console. Instead, each method produces ("returns") a numeric result.

The Math Class can be called through the following syntax:

Math.methodName(parameters)​

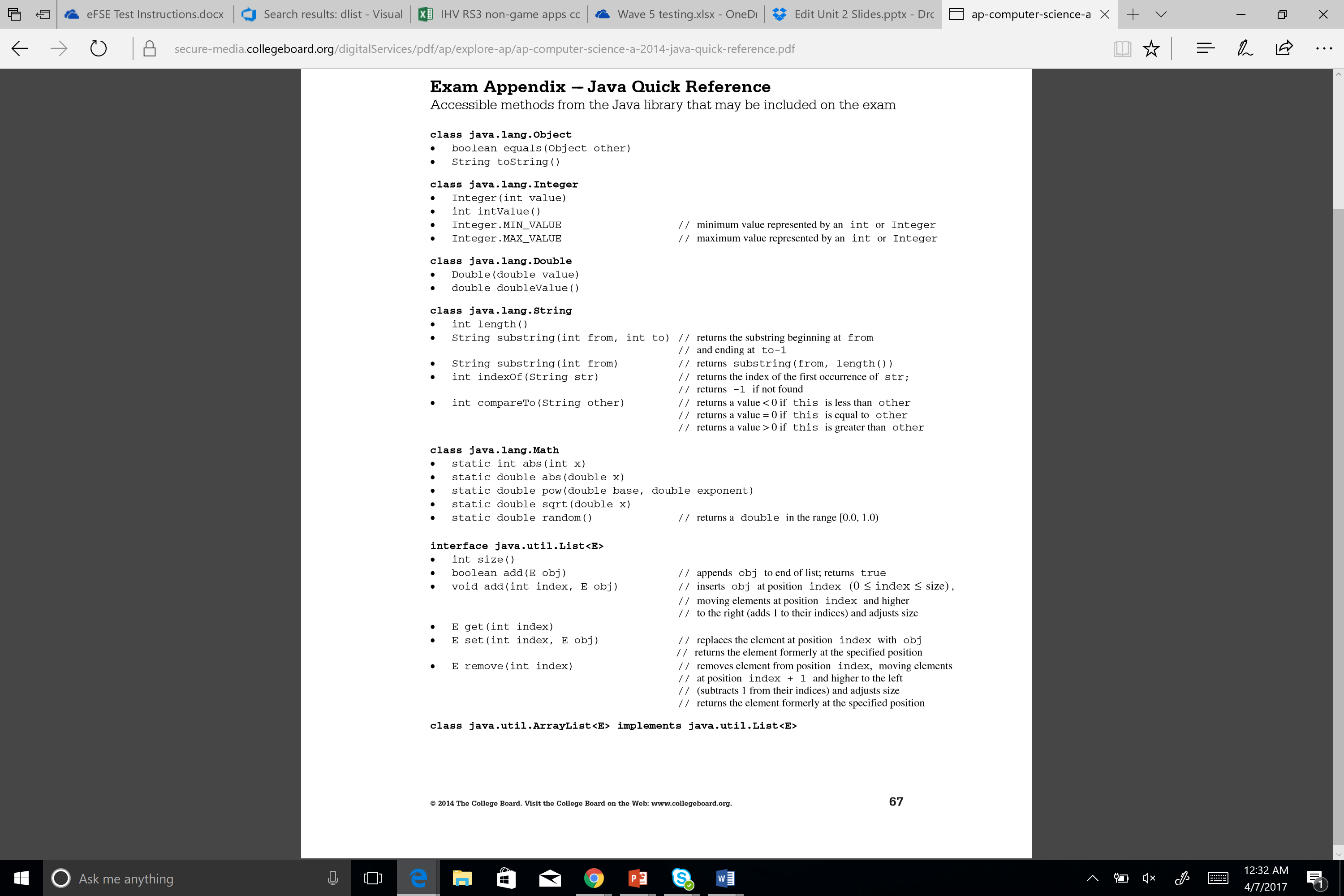
​

Example:​

​double squareRoot = Math.sqrt(121.0);​

System.out.println(squareRoot);           // 11.0​

These are the methods they will give you on the exam:



Strings

A Java string is a sequence of characters. They are objects of type String. Strings are Immutable, which means they cannot be changed after they’ve been created; the default constructor creates an empty string.​

Syntax

​String name = "text";​

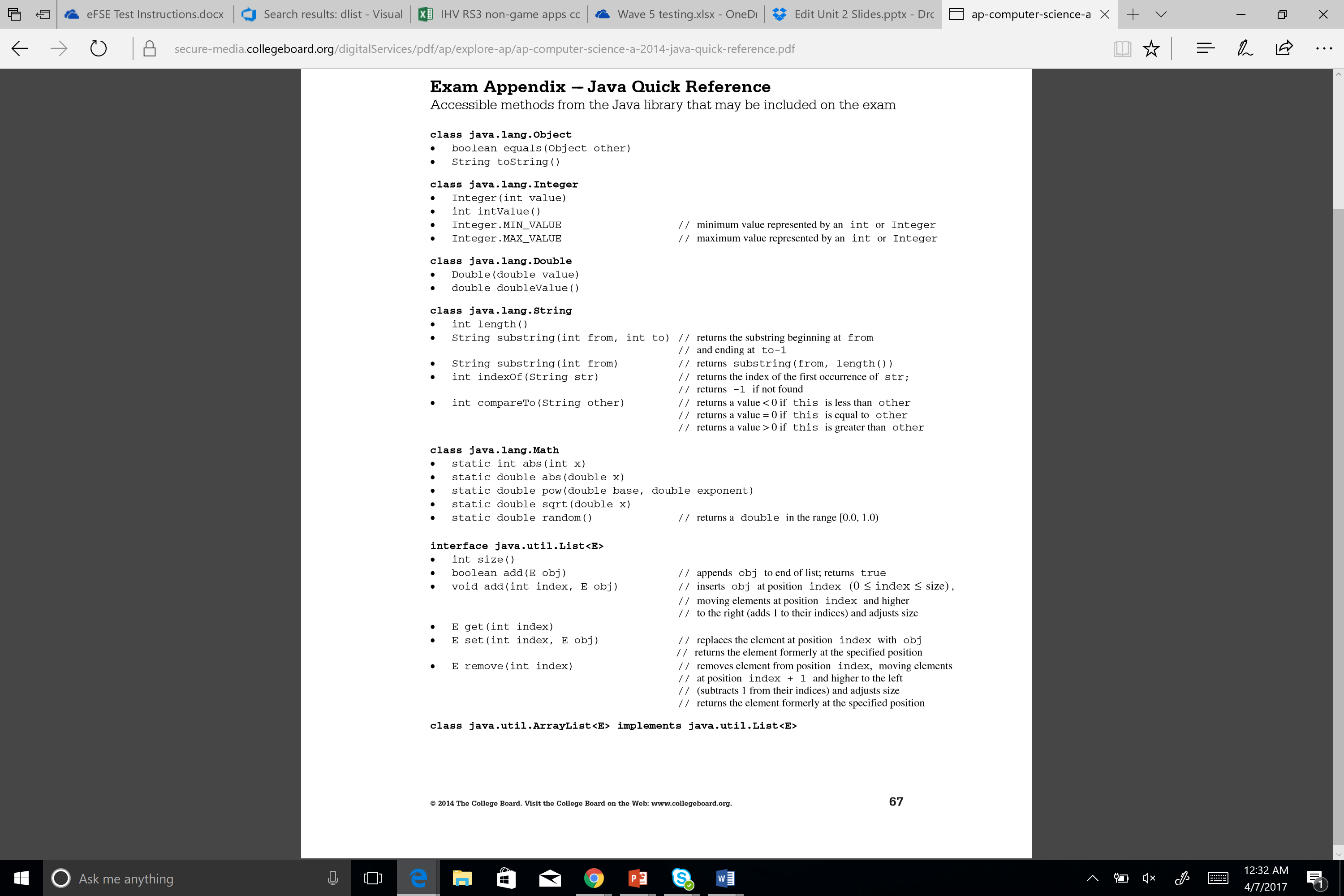
String name = expression;​

Examples:​

​String name = "Polly Roger";​

​String s = new String();

These are the methods they will give you on the exam:



Unit 2: Parameters and Returns - Problems

1. Consider the following code segment.

String str = “abcdef”;

for(int rep = 0; rep < str.length()-1; rep++){

System.out.print(str.substring(rep, rep+2));

}

What is printed as a result of executing this code segment?

1. abcdef
2. aabbccddeeff
3. abbccddeef
4. abcbcdcdedef
5. Nothing is printed because an IndexOutOfBoundsException is thrown

1. Assume that the following variable declarations have been made.

double d = Math.random();

double r;

Which of the following assigns a value to r from the uniform distribution over the range 0.5 <= 0 < 5.5?

1. r = d+0.5;
2. r = d+0.5\*5.0;
3. r = d\*5.0;
4. r = d\*5.0+0.5;
5. r = d\*5.5;

1. Consider the following method.

public String goAgain(String str, int index){

if(index >= str.length())

return str;

return str.goAgain(str.substring(index), index+1);

}

What is printed as a result of executing the following statement?

System.out.println(goAgain(“today”, 1));

(A) today

(B) todayto

(C) todayoday

(D) todayodayay

(E)todayodaydayayy

1. Which statement about parameters is false?
2. The scope of parameters is the method in which they are defined
3. Static methods have no implicit parameter this
4. Two overloaded methods in the same class must have parameters with different names
5. All parameters in Java are passed by value
6. Two different constructors in a given class can have the same number of parmeters
7. Consider these declarations:

String s1 = “crab”;

String s2 = new String(“crab”);

String s3 = s1;

Which expression involving these strings evaluates to true?

I s1 == s2

II s1.equals(s2)

III s3.equals(s2)

1. I only
2. II only
3. II and III only
4. I and II only
5. I, II, and III

6. Consider this method:

public static String doSomething(String s) {

   final String BLANK = “ ”; //String with one space

String str = “”; //empty string

String temp;

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

temp = s.substring(i, i+1);

if(!temp.equals(BLANK))

str+=temp;

}

Return str;

}

Which of the following is the most precise description of what doSomething does?

1. It returns s with all its blanks removed
2. It returns s unchanged
3. It returns a String that is equivalent to s with all its blanks removed
4. It returns a String that is an exact copy of s
5. It returns a String that contains s.length() blanks