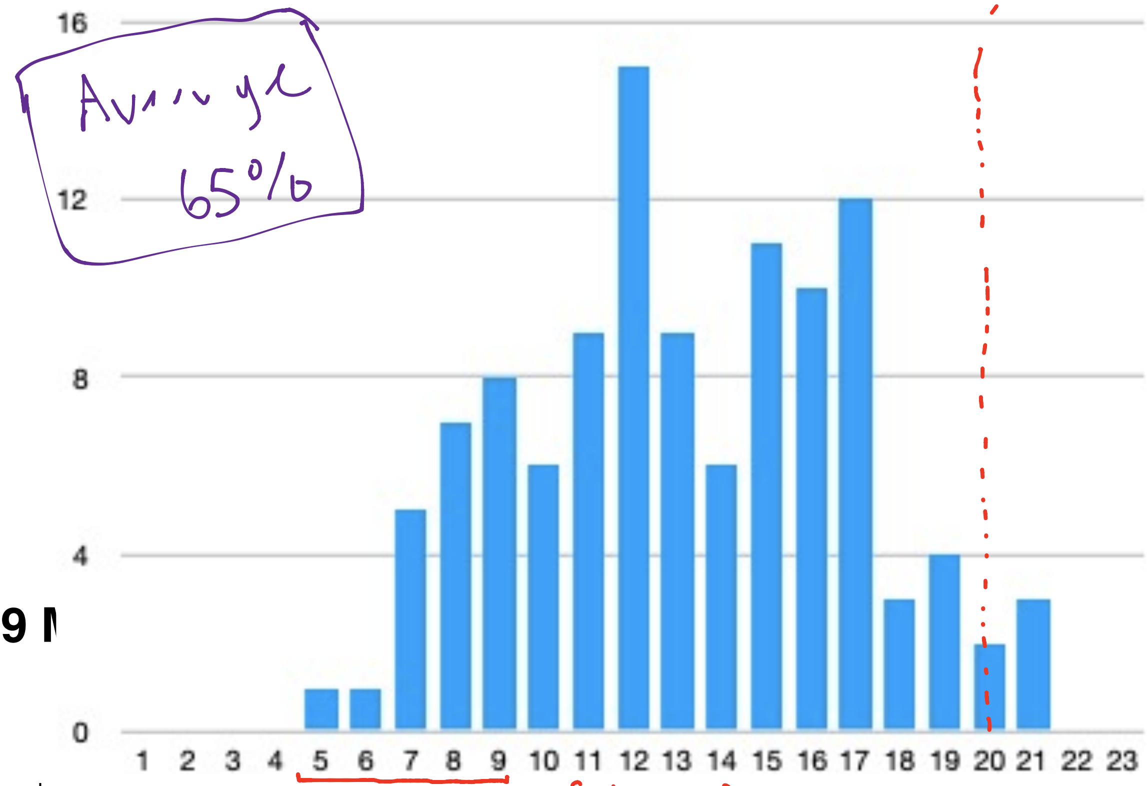
Page 2	Page 3	Page 4	Page 5	Total
4	8	5	5.5	22.5

Out of 20

LAST NAME	Marking Guide.
FIRST NAME	
STUDENT NUMBER	



waive failure if you pass the course

S

COMP 249 I

Winter 2019
Section S

Duration: 60 minutes
One single-sided letter-sized reference sheet of paper is allowed

Note on *negative marking*: it is a misconception that negative marking always punishes guessing. In this case, if you don't know the answer to a True/False question, you will expect 0.5 with 50% probability and -0.5 with 50%, which is an expected value of 0 — the exact same as if you left it blank. So go ahead and guess; you just won't be rewarded for guessing.

(4 Marks) A set of objects are defined as follows:

```
public class Parent {
 public Parent () {
  test();
 }
 public void test() {
  System.out.println(“01");
 }
}
```

```
class Child extends Parent {
 public Child () {
  test2();
 }
 public static void test2() {
   System.out.println(“03");
 }
 public void test3() {
   System.out.println(“04");
 }
}
```

super

test()

Print 03

```
class GrandChild extends Child {
 public GrandChild () {
  test3();
 }
 public void test() {System.out.println("05");}
 public static void test2() {System.out.println("06");}
 public void test3() {System.out.println(“07");}
}
```

super

test()

print 03

test 3()

For each line below, write the corresponding output or the note why the line is invalid.

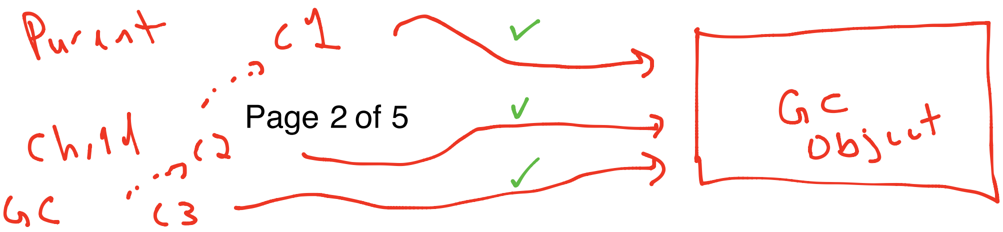
Code Snippet	*Output*
`Parent c1 = new GrandChild();`	05 / 03 / 07

`c1.test();`
`c1.test2();`
`c1.test3();`
	05 / err / err

`Child c2 = (Child)c1;` ✓
`c2.test();`
`c2.test2();`
`c2.test3();`
	05 / 03 / 07

`GrandChild c3 = (GrandChild)c2;`
`c3.test();`
`c3.test2();`
`c3.test3();`
	05 / 06 / 07

child / GC



(3 marks) Consider the following method to convert a Child into a GrandChild.

```
public GrandChild make(Child c) throws IllegalArgumentException {
 if (!(c instanceof GrandChild))
      throw new IllegalArgumentException("123");
 GrandChild gc = (GrandChild) c;
 return gc;
}
```

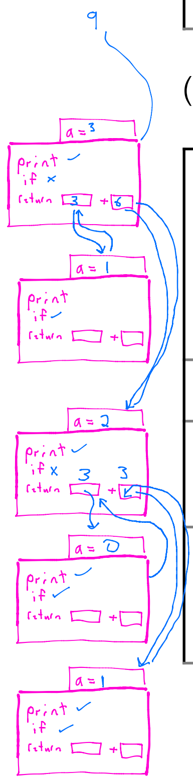
Specify for each if the code runs successfully, throws an exception, or is otherwise invalid.

`Parent p = new GrandChild();`
`GrandChild gc = make(p);`

Result: Invalid (parameter → wrong type)

`Child c = new GrandChild();` `GrandChild gc = make(c);` ✓
Result: Valid

`Child c = new Child();` `GrandChild gc = make(c);` ✓
Result: Exception



(5 Marks) Specify the output of the following recursive method.

```
public int Method(int a){
  System.out.print(a);
  if (a<2) return 3;
  return Method(a-2)+Method(a-1);
}
```

Code Snippet as we run	*Output*
`int result = Method(3);`	3 1 2 0 1
`System.out.println(result);`	9

returned int?

True/False Questions (0.5 mark for correct; -0.5 for incorrect; Minimum on all T/F is 0)

Exceptions A and B both directly extend RuntimeException. The order in which they are caught does not matter.	[✓ TRUE]	FALSE
Modify the third line of the recursive method on Page 3 with: `return Method(a-1)+Method(a-2);` Swapping the order like this will never change the return value.	[✓ TRUE]	FALSE
To append to a text file, the parameter TRUE is passed into the constructor of the PrintWriter (as opposed to FileOutputStream)	TRUE	[✓ FALSE]
To close a text file after writing to it, you close the PrintWriter (as opposed to the FileOutputStream)	[✓ TRUE]	FALSE
Generally, an infinite recursion will run longer before Java crashes due to memory use than an infinite loop will.	TRUE	[✓ FALSE]

method(a1, a2, a3)

Alice wants to create a new exception that will be unchecked. She should extend Exception (instead of RuntimeException)	TRUE	[✓ FALSE]
A static method in a parent method can be made non-static by the child method.	TRUE	[✓ FALSE]
`public int[] Method(int… a){return a;}` The above is valid code.	[✓ TRUE]	FALSE
A method is designated private in a parent class and overridden as public in the child class. An instance has the parent's variable type and the child's object type: the instance will run the parent's version of the method.	TRUE	[✓ FALSE]
ABC is an abstract class. A and B and C are all concrete classes that extend ABC. Instances of A, B, C can be stored in an array of variable type ABC.	[✓ TRUE]	FALSE

True/False Questions Continued

2 * 2 * 2 * 2
4 3 2 1
x 0

`public static int power(int x, int n) {`
`  if (n == 0) return 1;`
`  else return x * power(x, n-1);`
`}`
Given the above code, `power(2,4)` will return 16
	[✓ TRUE]	FALSE

The File object is slightly misnamed: it can in fact point at a directory (folder of files) as well as a file.	[✓ TRUE]	FALSE

The following is valid code (assuming java.io.* is imported):
`try{ PrintStream p = new PrintStream(`
`          new FileOutputStream(“test.dat”));`
`  boolean opened=true; }`
`catch(FileNotFoundException e){opened=false;}`
`finally{System.out.println(opened);}`
	TRUE	[✓ FALSE]

If the method `nextInt` is invoked on a Scanner object that is at the end of the text file, a `NoSuchElementException` will be thrown. This is a *checked* assumption.	TRUE	[✓ FALSE]
A parent class has a method that computers an average and returns the result as a double. A child class can override this method so that it returns a float instead.	TRUE	[✓ FALSE]

(3 marks) Write one (or more) catch block(s) that will catch all *checked* exceptions and will throw all *unchecked* exceptions. → RunTimeException

```
catch ( Exception e ) {
     if (e instanceof RuntimeException)
                    throw e;
}
```