Practice Final Exam

Program Design and Abstraction

Answer the questions in the spaces provided. **Please note** that there are no intentional errors in the code provided except in questions asking you to correct said code. Your written code does not have to be 100% syntactically correct.

Name: _____

Instructor: _____

Page	Points	Score
3	15	
4	14	
5	10	
6	15	
7	16	
8	15	
9	15	
Total:	100	

Please put your name on the top of every page.

Useful notes:

- You are allowed to clarify any answer you give.
- You are allowed to ask for clarification.
- Things are never as complicated as they appear, especially the math.
- Never leave a question blank, even if you don't know the answer. We can't give partial credit to blanks.
- `Math.pow(x,2)` returns the double x^2
- `Math.sqrt(x)` returns the double $\sqrt{x}$
- Extra credit is available for exceptional answers (up to five points).

Don't Panic

_____ 0 points

Short Answer

1. (5 points) What does the `static` keyword mean?

2. (5 points) Why and how do you use a `try`/`catch` block?

3. (5 points) What does the `void` keyword mean?

Code Evaluation

4. (9 points) What is the output of the following code block?

```
int d1 = 11;
int d4 = d1 % 2;
d1 /= 2;
int d3 = d1 % 2;
d1 /= 2;
int d2 = d1 % 2;
d1 /= 2;
System.out.println("Answer: " + d1 + " : " + d2 + " : " + d3 + " : " + d4);
```

5. (5 points) What is the error in the code?

```
boolean good = true;
while(good = true) {
//do stuff
}
```

Integer Methods

6. (5 points) `blackjack`: Given a pair of ints, return the int that is closest to 21 without going over.

```
// blackjack(7,17)  -> 17
// blackjack(21,16) -> 21
// blackjack(19,23) -> 19
public int blackjack(int a, int b) {









}
```

7. (5 points) `isDivisible`: Returns true if `x` is divisible by `y`.

```
// isDivisible(2,100) -> false
// isDivisible(4,2)   -> true
// isDivisible(123,3) -> true
public boolean isDivisible(int x, int y) {









}
```

String Methods

8. (5 points) `unOrUn`: If the String begins with "un", return a String with without the "un" in front. Otherwise, return the String with "un" added to the front of it. You may assume the String is at least 3 characters long.

```
// unOrUn("untied")    -> "tied"
// unOrUn("unable")    -> "able"
// unOrUn("necessary") -> "unnecessary"
public String unOrUn(String str) {









}
```

9. (10 points) `hasWildcat`: Given an input String `str`, return true if `str` contains the String "cat" in it, but the middle 'a' can be any `char`.

```
// hasWildcat("kitty")  -> false
// hasWildcat("tomcat") -> true
// hasWildcat("c4tn1P") -> true
public boolean hasWildcat(String str) {









} //  A-- would not buy again
```

 _____ 15 points

Array Methods

10. (10 points) `maxMinDiff`: Given an array of ints, return the difference between the maximum element and the minimum element. You can assume your array will have 2 or more elements in it.

```
// [1,2,3,4,5]  -> 4
// [15,31,21,17,28] -> 16
// [-1,-100,12,2,100] -> 200
public int maxMinDiff(int[] arr) {




}
```

11. (6 points) `swapEnds`: Given an array of ints, return the array with the first and last elements swapped. You can assume your array will have 2 or more elements in it.

```
// [1,2,3,4,5]  -> [5,2,3,4,1]
// [15,31,21,17,28] -> [28,31,21,17,15]
// [-1,-100,12,2,100] -> [100,-100,12,2,-1]
public int swapEnds(int[] arr) {




}
```


_____ 16 points

Project Euler Problems

12. (15 points) `euler1`: If we list all the multiples of 3 or 5 that are < 10, we get 3, 5, 6 and 9. The sum of these multiples is 23. `euler1` returns the sum of all multiples of 3 or 5 below `int limit`.

```
public int euler1(int limit) {




}
```

 _____ 15 points

13. (15 points) `euler2`: Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

$$1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...$$

By considering the terms in the Fibonacci sequence whose values do not exceed four million, write a program to find the sum of the even-valued terms.

```
public int euler2() {


}
```


_____ 15 points