

CSCE314: Programming Languages

Final Examination

May 5/6, 2016

Name: _____

UIN: _____

Section: _____ (501 Shell/500 Keyser)

Total time: 120 minutes

Total marks: 100, where 100 = 100%

Total total pages: 14

Question 1:	A _____	
	B _____	
	C _____	
	D _____	
		_____ / 30
Question 2		_____ / 25
Question 3		_____ / 8
Question 4		_____ / 7
Question 5		_____ / 22
Question 6:	A _____	
	B _____	
	C _____	
		_____ / 8
Total		_____ / 100

Plan and use your time wisely!

I have read and understand the Aggie Honor Code:

“An Aggie does not lie, cheat or steal or tolerate those who do.”

Signature: _____

Question 1: (30 points: please choose 3)

Question 1 is a Haskell question; all others are to be done in Java.

Choose 3 of the following 4 questions. Clearly indicate (*e.g.* by leaving completely blank, or crossing out) the one you choose not to answer. [10 points each; 30 points total]

Note: you may define helper functions for these programs as you deem appropriate. Also, you may use any of the routines listed in the following questions. The type signature for each is given:

```
getChar :: IO Char
putChar :: Char -> IO ()
return  :: a -> IO a
getLine :: IO String
putStr  :: String -> IO ()
putStrLn :: String -> IO ()
```

You also may use the standard parsing functions where you deem appropriate (*e.g.*, those in Hut-
ton, Ch. 8, or those in `Parsing.hs`).

Question 1–(A):

[10 points]

Write a parser that will parse expressions such as:

```
5==25
4==16
3==9
7==49
7==48
```

Your parser should return a `True` for the first 4 instances, and `False` for the final instance. Strings that match the form `'<x>==<y>'`, should return `True` if the value of x^2 (when x is interpreted as an integer) equals y (also an integer), and `False` otherwise. Strings that are of a different form (*e.g.*, `'7=='`, or as another example `'hello'`) should fail, (*i.e.*, return only an empty list).

Question 1–(B):

[10 points]

You're likely familiar with run-length encoding, or RLE, a common form of compression for data that are repetitious. Write a run-length decoder, `myRLD`, which will decode something encoded in this way. Here is an example of how the input is encoded (as a list of tuples), and the desired output.

```
Main> myRLD [(3,'h'),(1,'e'),(7,'l'),(1,'o'),(1,' '),(1,'w'),(1,'o'),(4,'r'),
              (1,'l'),(1,'d')]
"hhhellllllllo worrrrld"
```

Question 1–(C):

[10 points]

Write a routine, `getNames`, that asks a user for the number of people in a group, then asks for the name of each person, storing the names in a list that is returned. The type signature is:

```
getNames :: IO [String]
```

Question 1–(D):

[10 points]

You have developed a secret code system. Decoding messages is straightforward: every letter following a lower case 'e' is part of the hidden message, though "ee" counts as just one 'e'. So, a message like: "Her screen blinked and she kept thinking about weird Haskell else calls." would be "red pill". Write a Haskell function that reads in a string from a user and then outputs the decoded message.

Question 2: (25 points)

Imagine that you would like an interface to refer to various objects using both a formal name and a common name. Classes that allow this should allow a routine `setName (String)` that takes in and sets the informal name, a routine `formal ()` that returns the formal name, and a routine `informal ()` that returns the informal name. The following code should be valid:

```
public static void main(String[] args) {
    Royalty king = new Royalty();
    Lawyer defender = new Lawyer("Vinny");
    Judge districtjudge = new Judge("Judy");

    king.setName("Arthur");

    printgreeting(king);
    printgreeting(districtjudge);
    printgreeting(defender);
}
```

And would produce the following output when run:

```
Your Highness Arthur  
Your Honor Judy  
Attorney Vinny
```

You should create each of the following. Please label each part, and use a second page as necessary.

- (a) Define an appropriate interface for these. [5 points]
- (b) Define `Royalty`, `Lawyer`, and `Judge` classes, with `Judge` being a subtype of `Lawyer`. Note from the example code the way these are created and what formal name is output. [15 points]
- (c) Create a routine `printgreeting` that will print out the formal name followed by the informal. [5 points]

(Problem 2 — workspace cont.)

Question 3: (8 points)

Write a function that takes in the name of a class as a `String` parameter, and returns `True` if the class contains private attributes, and `False` otherwise. [8 points]

Question 4: (7 points)

Consider the following snippet of Java code:

```
public static void testFunc(ClassOne<? super ClassTwo, ? extends ClassTwo> x,  
                           ClassTwo y) {  
    . . .  
}
```

Suppose that `testFunc` is called with some code like this:

[7 points]

```
ClassI objectI = new ClassI();  
ClassJ objectJ = new ClassJ();  
ClassOne o1 = new ClassOne<ClassI, ClassJ>();  
ClassTwo o2 = new ObjectTwo();  
  
testFunc(o1, o2);
```

Given that this compiles and runs correctly, express the relationships between the following classes `Object`, `ClassI`, `ClassJ`, `ClassOne`, and `ClassTwo` using the subtype operator `:>`

Question 5: (22 points)

Imagine you are creating a game in which players repeatedly earn “trophy points”. There can be multiple players playing simultaneously (each player in a separate thread), and the trophy points are shared by all players. As the players collectively gain more trophy points, new events happen in the game. For every 1000 trophy points collected, there is a bonus generated. Code for some of the basic classes (not showing import commands or details) is on the next page. Clearly label your answers to the question on separate pages.

- (a) Show how you would write the `PlayerRunnable` class, such that in each thread, as long as the player is alive, we will get a player move, evaluate the result of that move, and award some number of trophies. [6 points]
- (b) Give an implementation of `awardTrophies` that will avoid race conditions. Your implementation of `awardTrophies` should increase the number of shared trophies by the amount passed in as a parameter. [5 points]
- (c) Create a game manager class, called `Manager`, that includes routines `awardBonus` and `increaseLevel`.
 - (i) Show the header for the `Manager` class. [3 points]
 - (ii) Show the run routine that must be created in `Manager` so that the `awardBonus` and `increaseLevel` methods are called at the appropriate point times. Note that this might require changes to the `awardTrophies` command in the `Player` class — you may show a modified `Player` class. [8 points]

```
public class Player {
    // Some attributes, constructors, methods, etc. defined here

    Player() { }

    public boolean isAlive() {
        //... This returns true if the player is alive
    }
    public Move getMove() {
        //... This gets a move from a user
    }
    public int getResult(Move nextmove) {
        //... This returns the number of trophies the move resulted in
    }
    public void awardTrophies(int numtrophies) {
        // PROBLEM B code will go here
    }
}
```

```

//THIS IS FOR THE HEADER LINE FOR THE Manager CLASS (Question C.i)
// Some attributes, constructors, methods, etc. defined here
Manager() { }

public void awardBonus() {
    // ... causes there to be some bonus activated in the game
}

public void run() {
    // CODE FOR PART C.ii GOES HERE
}
}

public class Main {
    // These can be accessed from other functions as Main.mylock,
    //           Main.cond, Main.trophies
    final public static Lock mylock = new ReentrantLock();
    final public static Condition cond = mylock.newCondition();
    public static int trophies;

    public static void main(String[] args) {
        // This is a 2-player situation
        Player p1 = new Player();
        Player p2 = new Player();
        // Stuff here to initialize players
        Manager m = new Manager();
        // Stuff here to initialize game manager

        PlayerRunnable pr1 = new PlayerRunnable(p1);
        PlayerRunnable pr2 = new PlayerRunnable(p2);
        Thread t1 = new Thread(pr1);
        Thread t2 = new Thread(pr1);

        t1.start();
        t2.start();

        m.start();
    }
}

```

(Problem 5 — workspace)

(Problem 5 — workspace cont.)

Question 6: (8 points: please choose 2)

Choose 2 of these 3 questions. Clearly indicate (*e.g.* by leaving completely blank, or crossing out) the one you choose not to answer. [4 points each; 8 points total]

For these questions, assume there are classes defined as follows, and objects with names will match the obvious type (*e.g.* `swivelchair1` is a `SwivelChair`, `furniture1` is a `Furniture`, *etc.*)

```
class Furniture;
class Chair extends Furniture;
class SwivelChair extends Chair;
class CaseGoods extends Furniture;
class Dresser extends CaseGoods;
```

(A) What is a function declaration for a function, `doSomething1`, that could be called in the following ways:

```
furniture1 = doSomething1(chair1, casegoods1);
casegoods1 = doSomething1(swivelchair1, furniture1);
```

(B) What is a function declaration, `doSomething2`, that could be called in the following ways:

```
SomePair<Chair, CaseGoods> thing1 = new SomePair<Chair, CaseGoods>;
SomePair<SwivelChair, Furniture> thing2 = new SomePair<SwivelChair, Furniture>;
doSomething2(thing1);
doSomething2(thing2);
```

(C) What are the types of the variables `thing3` and `thing4` in the following code:

```
thing3 = dresser3;  
thing4 = chair3;  
thing3 = casegoods3;  
thing4 = swivelchair3;
```