INFO1103: Introduction to Programming
Semester 2, 2014 - Study Quiz

Eric Liu

November 12, 2014

Week 1

Computer Basics

1. A computer consists of two types of ____ware?
2. What is "a supervisory program that oversees the operation of the computer" also called?
3. Name the type of language that computer hardware understands.
4. How do compilers and interpreters differ in translating code?
5. To what intermediate language does the Java compiler output?
6. What is the purpose of the Java Virtual Machine?
7. What is the difference between applications and applets?

Running Java Programs

1. What is the difference between the `java` and `javac` commands?
2. Where inside a Java program does execution begin?
3. If your Java class looks like the following:

```
public class Pikachu {
    public static void main(String[] args){
        System.out.print("Raichu");
        System.out.println("Zap");
    }
}
```

   What would this file be named, including file type?

4. If the above code is executed in the command-line, what happens?

Types of Errors

1. What is the difference between a run-time and compile-time error?
2. Consider the following code:

```
public class Pokemon {
    public static void main(String[] args){
        System.out.println("Go! Charmander!")
    }
}
```

   What happens when the program is run?

3. How can the above code be changed to run?

Week 2

Variables

1. What is the purpose of a variable?
2. List the eight primitive types in Java.
3. Write the statement which stores the value 25 inside an integer variable called `pikachu`.
4. Write the statement which updates `pikachu` to store the value of 100.
5. Write the statement which stores the first letter of the alphabet inside a character constant called `CHARMANDER`.

Arithmetic Operations

1. Consider the following code:

```
public class Hoenn {
    public static void main(String[] args){
        int cities = 10;
        int upgraded = 2 * (++cities);

        int towns = 10;
        int upgrading = 2 * (towns++);
        System.out.println(upgraded + " " + upgrading);
    }
}
```

   What is printed by the program?

2. What is the value of `towns` at the end of the program?
3. What is the value of a, if `a = 15%4 + 10%10` ?
4. What is the value of a, if `a = 10 % 0` ?
5. Write the statement that calculates 7^{20} and stores it in an integer variable called `magikarp`.

Casting

1. Write the statement that stores the constant π inside a float variable called `pie`.
2. What is the result of `5/2.0`?
3. What is the result of `5/2`?
4. What is the result of `(float)3/4`?

Keyboard Input/Screen Output

1. Write the statement which creates a Scanner object called `whatIs` that reads keyboard input, excluding import.
2. Write the statement which reads in a single word using the previous Scanner and stores it in a String variable called `thisWord`.
3. Consider the following code:

```java
public class PokeMart {
    public static void main(String[] args){
        double potionPrice = 2.5;
        // Print potion price to two decimal places here
    }
}
```

Write the statement which performs the above comment's function.

Strings

1. Write the statement which defines an empty String variable called `blank`.
2. Consider the following code:

```java
public class PokemonBattle {
    public static void main(String[] args){
        String wildPokemon = "Zubat";
        System.out.println("A wild " + wildPokemon.toUpperCase() + " appears!");
    }
}
```

What is printed by the program?

3. For the above program, what is the returned value of `wildPokemon.substring(1,3)`?
4. Write the statement that prints the phrase "true\false".

Documentation and Style

1. What is the difference between using `// comment` and `/* comment */` in Java?
2. What is the purpose of indentation?
3. What is the purpose of choosing descriptive variable names?

Boolean Expressions

1. List the two other logical operators besides `&&`.
2. List the five other boolean expressions besides `==`.
3. Consider the following code:

```java
public class Inequality {
    public static void main(String[] args){
        int x = 10;
        boolean result;
        // Store in result, the output of 0 < x < 10
        System.out.println(result);
    }
}
```

Write the statement which performs the above comment's function.

4. For the above program, what is printed?
5. What is the difference between `=` and `==`?

6. Why is `.equals()` used to compare equality of Strings as opposed to `==`?
7. What is the meaning of *lexicographic order*?

`if-else` Statements

1. Describe the syntax of a basic `if-else` statement.
2. Consider the following code:

```
public class NameOrdering {
    public static void main(String[] args){
        String itemOne = "Super Potion";
        String itemTwo = "Hyper Potion";

        if(/* condition */){
            System.out.println("Item one is before or equal to item two.");
        } else {
            System.out.println("Item two is before item one.");
        }
    }
}
```

   Write the condition that compares both items in lexicographic order for the result to be printed correctly.

3. Can the `else` component be omitted when writing `if-else` statements?
4. What is the purpose of nesting `if-else` statements?
5. What are `else-if` statements used for?

`switch` Statements

1. Describe the syntax of a `switch` statement with two cases and a default case.
2. Consider the following code:

```
public class FireTypeMatchup {
    public static void main(String[] args){
        String defendingType = "Water";

        switch(defendingType){
            case "Grass":
            case "Ice":
            case "Bug":
                System.out.println("It's super effective!");
                break;
            case "Fire":
            case "Water":
            case "Ground":
                System.out.println("It's not very effective.");
                break;
            default:
                System.out.println("It hit!");
        }
    }
}
```

What is printed by the program?

3. For the above program, what is printed if `defendingType` is "Normal"?

Week 3

`while` Loops

1. Describe the syntax of a `while` loop.
2. List in three or four steps, the actions of a `while` loop.
3. Consider the following code:

```
public class StepCount {
    public static void main(String[] args){
        int stepCount = 0;

        while(stepCount < 10){
            System.out.println(stepCount);
        }
    }
}
```

What will the above program print?

`do-while` Loops

1. Describe the syntax of a `do-while` loop.
2. List in three or four steps, the actions of a `do-while` loop.
3. What is the main difference between `while` and `do-while` loops?

`for` Loops

1. Describe the syntax of a `for` loop.
2. Which other loop is most similar to a `for` loop?
3. Write code that makes a `for` loop print "Meow" on new lines infinitely.
4. Consider the following code:

```
public class TrainerCard {
    public static void main(String[] args){
        for(int i = 0; i < 10; i++);{
            System.out.println("Pokemon!!");
        }
    }
}
```

What will the above code print?

5. Consider the following code:

```java
public class PokedexCount {
    public static void main(String[] args){
        for(int i = 0; i < 3; ){
            i++;
        }
        System.out.println(i);
    }
}
```

What will the above code print?

Terminating Loops

1. Describe the purpose of the `continue` keyword.
2. Consider the following code:

```java
public class Logic {
    public static void main(String[] args){
        int a = 0;
        int b = 1;
        int c = 2;
        int d = 3;

        // A
        while(a < b){
            // B
            while(c < d){
                break;
                c++;
            }
            // C
            a++;
        }
        // D
        System.out.println("Mew was here!");
    }
}
```

When the `break` statement is executed, which section of code is executed next? (A, B, C or D)

3. Write a program that requests a String input from the user, prints the inputted String and repeats until the word 'quit' is entered. Here is an example scenario:

```
Please enter a word: woof
You entered ... woof
Please enter a word: meow
You entered ... meow
Please enter a word: quit
Quitting
```

4. What is a sentinel value?

Loop Errors

1. List two ways to unintentionally create an infinite loop.

2. Write a `while` loop that prints the word "Pika" five times on the same line.

Week 4

Method Basics

1. Describe the syntax of a method excluding modifiers.
2. Consider the following code:

```
public class CatchPokemon {
    public static void main(String[] args) {
        if(throwPokeBall()){
            // A
            System.out.println("Gotcha! The Pokemon was caught!");
        } else {
            // B
            System.out.println("Oh no! The Pokemon broke free!");
        }
        // C
    }

    public static boolean throwPokeBall(){
        // D
        if(Math.random() > 0.5){
            return true;
        } else {
            return false;
        }
    }
}
```

   Which section of code is executed after `throwPokeBall()` is called from the `main` method? (A, B, C or D)

3. For the above program, if `throwPokeBall()` returns false and finishes executing, what is printed?
4. What does the `void` return type signify?
5. What is the purpose of creating methods?

Passing Parameters

1. Write a method with the following signature `public static boolean isOdd(int n)`, it should return `true` if the passed parameter is an odd number, `false` otherwise.
2. Consider the following code:

```
public class PokemonBattle {
    public static void main(String[] args) {
        boolean isBattle = true;
        String currentPokemon = "Gardevoir";

        if(isBattle){
            activateMega(currentPokemon);
        }
        System.out.println(currentPokemon);
    }
```

```
    public static void activateMega(String pokemon){
        pokemon = "Mega " + pokemon;
    }
}
```

When the above program is run, what is printed?

3. Write a program that requests the user to input an integer and then prints the square of the value. Include and use the following method signature in your program: `public static int square(int n)`. The program should loop until a negative value is entered.
Below is an example scenario:

```
Please enter an integer: 5
Square is ... 25
Please enter an integer: 10
Square is ... 100
Please enter an integer: -2
Quitting
```

Variable Scope

1. Consider the following code:

```
public class SilphCo {
    int x = 10;

    public static void main(String[] args) {
        int y = 2;
        for(int z1 = 5; z1 < 10; z1++){
            // A
        }
        // B
    }

    public static void useSilphScope(){
        for(int z2 = 1; z2 < 100; z2++){
            // C
        }
        // D
    }
}
```

List the sections where variable `x` can be seen. (A, B, C and/or D)

2. Similarly, for the above program, list where `y` can be seen.

3. Similarly, list where `z1` can be seen.

4. Similarly, list where `z2` can be seen.

Week 5

Array Basics

1. Describe the syntax of creating an array.

2. What is an array variable actually referring to?
3. Write the statement(s) that creates a double array storing two values: 0.21 and 1.31.
4. Consider the following code:

```
public class PokemonParty {
    public static void main(String[] args) {
        String[] pokemonList;
        pokemonList[0] = "Bulbasaur";
        System.out.println(pokemonList[0]);
    }
}
```

   What is printed by the above program?

5. How can the above program be made to work?

Common Array Algorithms

1. Consider the following code:

```
public class PokedexEntry {
    public static void main(String[] args) {
        String[] latios = new String[4];
        latios[0] = "Latios";
        latios[1] = "Type: Dragon/Psychic";
        latios[2] = "Species: Eon";
        latios[3] = "Description: Such fast! Much wow!";
        printData(latios);
    }

    public static void printEntry(String[] entry){
        // Print each element of array on a new line
    }
}
```

   Write the code that performs the function as specified in the comment above.

2. Consider the following code:

```
public class Search {
    public static void main(String[] args) {
        int[] itemQuantities = {11, 21, 8, 10}

        System.out.println("Smallest quantity is: " + findMin(itemQuantities));
    }

    public static int findMin(int[] array){
        // Find and return smallest value in array
    }
}
```

   Write the code that performs the above comment's function.

2D Arrays

1. Consider the following code:

```
public class OverworldTravel {
    public static void main(String[] args) {
        int[][] grid = new int[3][2];

        for(int i = 0; i < grid.length; i++){
            for(int j = 0; j <= grid[i].length; j++){
                System.out.print(grid[i][j]);
            }
            System.out.println();
        }
    }
}
```

What will be printed when the above program is run?

2. What should be printed if the error is fixed?

Week 6

More 2D Arrays

1. Consider the following code:

```
public class PokemonStatistics {
    public static void main(String[] args) {
        int[][] rawStats = {{5, 4, 5},
                            {1, 2},
                            {10, 11, 4}};

        int[] summedStats = summate(rawStats);

        for(int i = 0; i < summedStats.length; i++){
            System.out.println(summedStats[i]);
        }
    }

    public static int[] summate(int[][] data){
        // Calculate the sum of each row of data and store it in a new array
    }
}
```

Write the code the performs the above comment's function. Below is the expected output of the program:

```
14
3
25
```

More Array Algorithms

1. Consider the following code:

```
public class PokemonBox {
    static int currentSize = 4;
    static String[] currentBox = {"Charmander",
                                  "Bulbasaur",
```

```
                                    "Pikachu",
                                    "Squirtle" };

    public static void main(String[] args){
        release("Bulbasaur");
        for(int i = 0; i < currentSize; i++){
            System.out.println(currentBox[i]);
        }
    }

    public static void release(String name){
        // Remove the given name from the global array and update the size
    }
}
```

Write the code the performs the above comment's function. Below is the expected output of the program:

```
Charmander
Pikachu
Squirtle
```

2. Consider the following code:

```
public class PokeMartStock {
    public static void main(String[] args){
        int[] quantities = {10, 18, 20, 31, 5, 1, 2};
    }
}
```

Describe a simple solution to sort the above array into ascending order.

3. What is the difference between a linear search and a binary search?

The `ArrayList`

1. Describe the syntax for creating an `ArrayList`.
2. What is the most convenient aspect of using an `ArrayList` over an array?
3. List the statements which do the following on an `ArrayList`:
   (a) get the element at index i
   (b) remove the element at start of list
   (c) get the current size of list
   (d) add an element to end of list
4. Describe how to create a copy of an `ArrayList`.
5. What is a "Wrapper" used for in an `ArrayList`.
6. Write a program that requests integers from a user and stores them in an `ArrayList`. Then find and print the smallest integer in the `ArrayList`. An example scenario is shown below:

```
Please enter integers below or 'quit' to stop:
5
6
```

```
-1
2
quit
The smallest integer is -2
```

Week 7

Class and Object Basics

1. What is Object-Oriented Programming about?
2. Describe the relationship between objects and classes.
3. Give a real-life example of a class and object relationship.
4. How are object variables stored in comparison to primitive variables?
5. Write code to create a class called `Item` and declare three *instance variables*, a String `name`, a String `description` and an integer `quantity`.

Constructors

1. Describe the syntax of a constructor.
2. Describe the purpose of a constructor.
3. Describe the purpose of a constructor such that a 5-year old child can understand it. (This is serious)
4. Describe the syntax of a constructor one more time. (Also very serious)
5. Consider the following code:

```
public class Pikachu {
    String name;
    int HP, ATK, SPA, DEF, SPD, SPE;

    // Add a constructor here
}
```

   Write the constructor that initializes the default Pikachu object to the following values:

   (a) name is "PIKACHU"
   (b) HP is 10
   (c) ATK is 8
   (d) SPA is 14
   (e) DEF is 6
   (f) SPD is 7
   (g) SPE is 20

6. Consider the following code:

```
public class Player {
    String name;

    public Player(String name) {
        name = name;
    }
}
```

Describe how the above program can be fixed.

Week 8

Static and Non-static

1. Describe the purpose of static and non-static *variables*.
2. Consider the following phrase: "Every `Person` object has a `name`."
   Should `name` be a static or non-static variable and why?
3. Describe the purpose of static and non-static *methods*.
4. Consider the following phrase: "Every `Pikachu` object can `talk()` by printing out 'Pika!'"
   Should `talk()` be a static or non-static method and why?
5. What is the difference between a "class variable" and an "instance variable"?
6. What is the difference between a "class variable" and a "class type variable"?
7. What is a static method/variable also known as?
8. What is a non-static method/variable also known as?
9. What is the rule regarding the types of variables *static methods* can refer to and *non-static methods* can refer to?

Encapsulation

1. Explain why encapsulation is important.
2. What is the difference between `public` and `private` variables?
3. Consider the following code:

```
public class MayProfile {
    public String name = "May";

    public String getName(){
        return this.name;
    }
}
```

   Why is the above code problematic in regards to encapsulation?

4. What is the difference between a class interface and a class implementation?

More Constructors and Methods

1. Consider the following code:

```
public class TrainerCard {
    private int id;

    public TrainerCard(){
        id = 111;
    }

    public TrainerCard(int id) {
```

```
        this.id = id;
    }
}
```

What is the purpose of having multiple constructors?

2. Consider the following code:

```
public class Zigzagoon {
    public int id;
    public String name;
    public boolean isWild;
}
```

What does the constructor for the above class do? (Important question)

3. Write code that will generate a `NullPointerException`.

4. Consider the following code:

```
public class PokemonMath {
    public int sum(int a, int b){
        // returns result as an integer
    }

    public int sum(int a, int b, int c){
        // returns result as an integer
    }

    public String sum(int a, int b){
        // returns result as a String
    }
}
```

What is the problem with the above code?

5. Explain how `==` differs from `.equals()`.

Week 9

More Arrays and ArrayLists

1. Consider the following code:

```
public class AccountTester {
    public static void main(String[] args){
        ArrayList<Account> list = new ArrayList<Account>();
        Account[] array = new Account[10];

        Account kitty = new Account("Meow", "burgers");
        Account puppy = new Account("Woof", "cupcakes");

        // Add both accounts to the ArrayList and array
        // Then, print kitty's name and secret from the ArrayList
        // Also, print puppy's name and secret from the array
    }
}
```

```
class Account {
    public String name;
    private String secret;

    public Account(String name, String secret){
        this.name = name;
        this.secret = secret;
    }

    public String getSecret(){
        return this.secret;
    }
}
```

Write the code that performs the above comment's function. Below is the expected output:

```
Meow likes burgers.
Woof likes cupcakes.
```

Inheritance

1. Describe the benefits of using inheritance.
2. Give a real-life example of inheritance.
3. What is meant by a subclass and superclass?
4. Describe the syntax for using inheritance.
5. What is the purpose of the `protected` modifier?
6. Consider the following code:

```
class Bulbasaur {
    public int level;
    public String name, type;
    public int[] stats;

    public void useTackle(){
        System.out.println("It did 8 damage!");
    }
}

class Charmander {
    public int level;
    public String name, type;
    public int[] stats;

    public void useScratch(){
        System.out.println("It did 10 damage!");
    }
}
```

Describe a solution using inheritance to reduce the lines of code within the `Bulbasaur` and `Charmander` classes.

7. How can you call the constructor of a superclass class?

Method Overriding

1. What is the difference between *method overriding* and *method overloading*?
2. Give a real-life example of method overriding.
3. Describe how you give a method the power to override its parent.
4. Consider the following code below:

```
class Animal {
    public void talk(){
        System.out.println("Grrr...");
    }
}

class Dog extends Animal {
    // Add talk method here
}

public class AnimalTester {
    public static void main(String[] args){
        Animal wildAnimal = new Animal();
        Animal friendlyAnimal = new Dog();
        wildAnimal.talk();
        friendlyAnimal.talk();
    }
}
```

Write code that performs the above comment's function. Below is the expected output:

```
Grrr...
Bark!
```

Polymorphism

1. What is the benefit of using polymorphism?
2. Is there a syntax for polymorphism, why?
3. Consider the following code:

```
class Animal {
    public void talk(){
        System.out.println("Wrrnngh");
    }
}

class Dog extends Animal {
    @Override
    public void talk(){
        System.out.println("Woof!");
    }
}

class Cat extends Animal {
    @Override
    public void talk(){
        System.out.println("Meow!");
```

```
    }
}

public class PolymorphismTester {
    public static void main(String[] args){
        // Write code here that demonstrates polymorphism with the above classes
    }
}
```

You can do it! Don't give up!

4. What is the difference between a static-typing of objects and dynamic-typing of objects?
5. What mechanism does polymorphism use to achieve its features?
6. What is the purpose of the `instanceof` operator?

The `Object` Class

1. In your own words, describe what is so special about the `Object` class.
2. Give a real-life example of the `Object` class. (Possibly difficult)

Week 10

Mid-Semester Break

Pokemon time.

Week 11

Java Interfaces

1. Describe the syntax for implementing an interface.
2. In your own words, what real-life example would you relate an interface type to?
3. Describe the differences between a class interface, graphical user interface (GUI) and interface type.
4. Write an interface called `Talkable` that contains a method with the signature `void talk()`.
5. Write a class called `Animal` that implements the `Talkable` interface. The `talk()` method should print "Raawr" when called.
6. What are the differences between a class and an interface type.
7. What are the benefits of using an interface?
8. What is the benefit of extending an interface with another interface?
9. Describe how the `.compareTo()` function works in the `Comparable` interface.

`for-each` Loop

1. Describe the syntax of a `for-each` loop.
2. What is the benefit of using a `for-each` loop?
3. What are the limitations of a `for-each` loop?
4. Consider the following code:

```
public class PokeCentre {
    public static void main(String[] args){
        ArrayList<Pokemon> party = new ArrayList<Pokemon>(6);
        // 6 Pokemon objects are added to party

        for(int i = 0; i < party.size(); i++){
            Pokemon current = party.get(i);
            current.heal();
        }
    }
}

class Pokemon {
    public int maxHP = 100;
    public int HP = 20;

    public void heal(){
        HP = maxHP;
    }
}
```

Replace the `for` loop with a `for-each` loop that performs the same function.

Abstract Classes

1. What is the purpose of abstract classes?
2. Give a real-life example of an abstract class.
3. What are the differences between abstract classes and interfaces?
4. Consider the following code:

```
class PlayerCharacter {
    private int x, y;

    public void move(int newX, int newY){
        this.x = newX;
        this.y = newY;
    }

    public int getX(){
        return this.x;
    }

    public int getX(){
        return this.y;
    }
}
```

```
class NonPlayerCharacter {
    private int x, y;
    private int pushback = 5;

    public void move(int newX, int newY){
         this.x = newX - pushback;
         this.y = newY - pushback;
    }

    public int getX(){
        return this.x;
    }

    public int getX(){
        return this.y;
    }
}
```

Write an abstract class called `Character` such that extending this abstract class removes redundant code and emphasises the derived nature of the above two classes.

Week 12

Reading/Writing Files

1. Describe how to set up a `Scanner` for reading text files.
2. When you have finished editing the text file, what must you not forget to do?
3. When opening a file, which exception has a chance to be thrown?
4. Write a program that requests the user to enter a sentence, then output a text file called "awesome.txt" which contains the input sentence.
   Below is an example scenario:

   ```
   Please enter a sentence:
   Rayquaza wears lipstick!
   ```

   A text file called "awesome.txt" should be created and contain the above sentence.
5. Describe how to *append data* as opposed to overwrite data when using a `PrintWriter` object.

Processing Text

1. What is the purpose of changing a `Scanner` object's delimiter to another String *besides white space*?
2. List five useful methods, in your opinion, for manipulating Strings.
3. List five useful methods, in your opinion, for working with characters. (The `Character` class is handy)
4. Consider the following text:

   ```
   Brock Rock
   Misty Water
   Lt.Surge Electric
   Erika Grass
   Koga Poison
   Sabrina Psychic
   ```

```
Blaine Fire
Giovanni Ground
```

Write a program that prints the above text in a more descriptive format. The file is named "data.txt" and is located in the root directory. Below is the expected output:

```
Brock is the Rock-type Gym Leader.
Misty is the Water-type Gym Leader.
Lt.Surge is the Electric-type Gym Leader.
Erika is the Grass-type Gym Leader.
Koga is the Poison-type Gym Leader.
Sabrina is the Psychic-type Gym Leader.
Blaine is the Fire-type Gym Leader.
Giovanni is the Ground-type Gym Leader.
```

Command-line Arguments

1. Describe the syntax for passing arguments through the command-line.
2. Where are the command-line arguments stored in a program?
3. Describe how to access passed command-line arguments.

Exceptions and Exception Handling

1. What is the purpose of an `Exception`?
2. Describe the syntax of a `try-catch` block.
3. Describe the syntax of a `throws` statement.
4. Consider the following code:

```
public class Glitch {
    public static void main(String[] args){
        int pokedexNo = 0;
        if(pokedexNo > 0){
            System.out.println("Wild Pokemon!");
        } else {
            throw new Exception("Missingno!");
        }
    }
}
```

Write the above code again after modification in the following two scenarios:

(a) Handle the `Exception` with a `try-catch`

(b) Handle the `Exception` with a `throws`

5. List five common predefined `Exception` objects you have come across. (E.g. `NullPointerException`)
6. What happens if an `Exception` is not caught and the program is executed?
7. Explain why more specific `Exception` objects should be caught before more general ones.

Week 13

Recursion

1. What are the advantages using of recursion?
2. What are the disadvantages of using recursion?
3. List the two requirements for any recursive method.
4. What happens when an infinitely recursive method is executed?
5. Write a method that recursively computes a factorial and uses the following method signature: `public static int factorial(int n)`. (Obligatory example)
6. Write a method that *iteratively* computes a factorial, using the same method signature as above.
7. Consider the following information:

```
Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

fibonacci(0) -> 0
fibonacci(1) -> 1
fibonacci(2) -> 1
```

   Fibonacci numbers defined by the sum of the two previous values (starting from 0 and 1, base cases). Assuming n is always positive, write a method which determines the nth Fibonacci number recursively. Use the following method signature: `public static int fibonacci(int n)`

Credits and Thanks to:

- Irena Koprinska (Lecturer) - For making this awesome programming course!
- Darren Shen (Tutor) - For tons of advice and recommendations for questions!
- Chris Chen (Student) - For revision notes and recommendations for questions!
- Pokemon - Because Pokemon! :D

> Some people want it to happen,
> Some wish it would happen,
> Others make it happen.
>
> *Michael Jordan*