CP122 Placement Test Study Guide

1 Variables and Types [10 pts]

You should know how to declare, initialize, and manipulate standard data types. You should know the difference between primitive types vs. objects, and the effects thereof. You should know how the type system handles interactions between variables with similar but distinct (e.g. int vs. double) types.

2 Operators and Methods [10 pts]

You should know how to use operators and methods to manipulate variables and perform computations. You should know the syntax and use cases of the most common methods associated with common data types.

3 Control Flow [10 pts]

You should know how to use conditional statements, for loops, and while loops to control the flow of a program's execution. You should be able to trace the execution of a program using multiple control flow structures, and reason about the overall behavior of such programs.

Example question(s):

```
// Which possible values for a, b, and c will execute the println statement?
int a = ?, b = ?, c = ?;
if (a > b || a > c) {
    System.out.println("Success!");

// What is the result of running the following code snippet?
int[] numbers = {1, 2, 3};
int total = 0;
for (int number : numbers) {
    for (int i=0; i < number; i++) {
        total += number;
    }
}
System.out.println(total);
```

4 Scope and Mutability [15 pts]

You should know how Java's scoping rules control access to and the values of variables.You should know how data is updated via assignment and mutation, and how this interacts with the type system.

Example question(s):

```
// What is the result of running the following code snippet?
int x = 13;
if (x > 15) {
    boolean large = true;
} else {
    boolean large = false;
}
System.out.println(large);

// What is the result of running the following code snippet?
public static void triple(int a) {
    a = 3*a;
}
public static void testTriple() {
    int x = 37;
    triple(x);
    System.out.println(x);
}
```

5 Functions [15 pts]

You should know how functions are declared and implemented. You should know how data is passed into, and retrieved from, functions. You should be able to trace the execution of a stack of function calls.

Example question(s):

```
// What is the result of running the following code snippet?
public int f(int x) {
    return x*3;
}

public int g(int x, int y) {
    return f(y) + x;
}

int x = 3;
int y = 5;
System.out.println(g(x, f(y)));
```

6 Classes and Interfaces [20 pts]

You should know how and why classes are used to organize and execute Java programs. You should know how classes are defined using attributes and methods, and how modifiers are used to control external access. You should know how objects are created and used. You should know how and why interfaces are used.

Example question(s):

```
// Which of the following describes the relationship between classes and objects?
// ...

// Which of the following is true about a class that implements an interface?
// ...

// What is the result of running the following code snippet?
class Cat {
    String name = "Mittens";
}
Cat izzy = new Cat();
Cat zoe = new Cat();
System.out.println("izzy and zoe are identical: " + (izzy == zoe));
System.out.println("izzy and zoe are equal: " + (izzy.equals(zoe)));
```

7 Object Oriented Programming [20 pts]

You should know how the core principles of Object Oriented Programming are used to inform best practices for writing programs in Java. You should know how and when to implement subclasses, and the interactions between a subclass and superclass. You should know how modifiers are used to control relationships between classes. You should know how Java's type system interacts with inheritance and polymorphism.

Example question(s):

```
// Which of the following best describes an advantage of
// encapsulation in object-oriented programming?
// ...

// Which of the following best describes a use for an abstract class?
// ...

// Which of the following statements about overriding methods are true?
// ...
```

```
// What is the result of running the following code snippet?
class Animal {
    void speak() { System.out.println("[DEFAULT ANIMAL NOISE]"); }
}

class Dog extends Animal {
    void speak() { System.out.println("Dog"); }
}

public class Test {
    public static void main(String[] args) {
        Animal a = new Dog();
        a.speak();
    }
}

// What is the result of running the following code snippet?
class Vehicle {
    Vehicle() { System.out.println("Base"); }
}
class Car extends Vehicle {
    Car() { System.out.println("Derived"); }
}
public class Main {
    public static void main(String[] args) {
        Car myRide = new Car();
    }
}
```