ESE112– Programming with Java

Quiz 1 — September 28, 2006

Name: ________________________________

Student ID: ________________________________
(from your PennCard)

Email ________________________________ @seas.upenn.edu

Instructions:

• You have 50 minutes to answer all of the questions. The quiz is worth 75 points. The point value of each question is given.

• Write your answers on the quiz pages. The back side of each page may be used as a scratch pad.

• Questions during the quiz should be about the wording only. If you have a question, raise your hand and we’ll come to you. (This is less disruptive for others than if you come to us.)

• DON’T PANIC! If you find a question that you cannot solve right away, consider moving on and returning to it after you finish the rest of the questions.

• Good luck!

<table>
<thead>
<tr>
<th>Score</th>
<th>Max</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>variables, operators, etc.</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>variables, operators, etc.</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>if statements</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>while loops</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>basics of OOP</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>writing code for a Car class</td>
</tr>
<tr>
<td>Total</td>
<td>75</td>
<td></td>
</tr>
</tbody>
</table>
Programming Language Basics: variables, operators, etc.

1. Give an example of an expression and an example of a statement. Then describe the difference between an expression and a statement in the general case. 4 points

2. Give the value of the variable x at the end of each set of interactions below. (You can assume that the RESET button in DrJava is hit after each set of interactions.

(a) > int x = 20;
    > int y = 50;
    > x = x / 2 + y;

(b) > boolean x = (5 > 3 || 3 < 2)
    > x = !x;

(c) > int num = 10;
    > int val = 3;
    > double x = (double)(num/val);

(d) > char c1 = 'm';
    > char c2 = 'x';
    > char x = c1;
    > c1 = c2;
    > c2 = x;
    > x = c1;

(e) > // assume x is declared appropriately
    > x = ((10 % 3) == 2);
if statements

3. Complete the methods isAChild/Teenager/Adult. Anyone younger than 13 is considered to be a child. An adult is older than a teenager.

Sample Interactions:

```java
> Person p1 = new Person(12);
> p1.isAChild()
true
> p1.isATeenager()
false
> p1.isAnAdult()
false
> p1 = new Person(19);
> p1.isAChild()
false
> p1.isATeenager()
true
> p1.isAnAdult()
false
```

```java
/* Person class */
public class Person{
    private int age;

    public Person(int years){ age = years; }
    public int getAge(){ return age; }

    public boolean isAChild(){

    }

    public boolean isATeenager(){

    }

    public boolean isAnAdult(){

    }
}
```
while loops

4. Complete this method:

```java
/*
 * Returns the sum of numbers from 1 to num. Returns 0 if num is not positive.
 */
public int sumOneTo(int num){
```
Basics of OOP

5. (a) Briefly, what does ”state of an object” mean?  

(b) When a new object is created with the ”new” operator, memory is allocated for the object and a named piece of code is executed which is called what?  

(c) True or False: When the state of an object is changed, the state of all the objects with the same type change.  

(d) True or False: A class is a template for creating objects.  

(e) True or False: A method whose return type is not void must have one or more return statements.  

(f) Consider an expression which is a method call, e.g. ”thing.doit()” . What is the ”type” of the expression?
Write code for a Car class on the next (blank) page. Upon creation, all Cars have an empty gas tank and an odometer reading of 0 (no miles driven). Cars can be created in two ways: with a default constructor which sets the miles per gallon to be 30, or with another constructor in which the miles per gallon is supplied as input (if the input is not positive, the miles per gallon is set to 30).

Gas can be added to a Car's tank with the addGasGallons() methods. Any positive number of gallons may be added to the tank. Inputs of negative gas gallons are ignored (no action taken). A Car can be driven with the drive() method which takes a number of miles as input. If it has enough gas to drive all of the miles requested, it should do so, update the state accordingly (gas tank and odometer), and return true. Otherwise it should return false. The following interactions should work.

```
// Test the constructors
> Car c1 = new Car();
> c1.getMilesPerGallon()
30
> Car c2 = new Car(45);
> c2.getMilesPerGallon()
45
> Car c3 = new Car(-20);
> c3.getMilesPerGallon()
30
> c1.getMilesDriven()
0.0
> c1.getGasGallons()
0.0
> // Now add some gas and drive!
> c1.addGasGallons(10)  // ok to make this method return void
> c1.drive(60)
true
> c1.getMilesDriven()
60.0
> c1.getGasGallons()
8.0
> c1.drive(210)
true
> c1.getMilesDriven()
270.0
> c1.getGasGallons()
1.0
> c1.drive(40)
false
> c1.getMilesDriven()
270.0
> c1.getGasGallons()
1.0
```
Code for a Car class