CSE-1020 Final Exam
Written Portion

Family Name: ______________________________
Given Name: ______________________________
Student#: ______________________________
CSE Account: ______________________________
Section: **A** **E** (Circle which section you are in.)

Instructors: Parke Godfrey (§A) & Burton Ma (§E)
Exam Duration: 80 minutes
Term: Fall 2009

Instructions

1. Write your answers clearly and succinctly.
   (a) You will receive no point for an unclear answer.
   (b) There is no additional deduction for a wrong answer.
   (c) Do not use red ink.
2. No aids (such as calculator, reference sheets, textbooks, etc.) are permitted.
3. Please turn off cell phones, and put your cell phones off the desk.
4. Generally, no questions regarding the interpretation, intention, or further elucidation of an exam question will be answered by invigilators.
   If truly in doubt, state your interpretation next to your answer.
5. For any of the program code, assume that
   (a) any program fragment is properly within a `main` method within a `class`,
   (b) any needed classes have been imported,
   (c) any constructor call has a legitimate matching constructor, and
   (d) that any unseen part of the program is correct.
   Thus, it would compile and run, *except perhaps* because of the program fragment shown.

Grading Box	
1.	/10
2.	/10
3.	/10
4.	/10
5.	/10
Total	/50

1. **Language Constructs.** [10pts]

   Each question (a)–(j) refers to a line of code in the API and `main` from page 3. Match that line from the code to the concept on the right (by its letter, **A**–**L**) that it *best* illustrates.

   Note that no concept, **A**–**L**, is a best answer more than once. Also note that there are *twelve* concepts, **A**–**L**, but just ten questions, (a)–(j). So two concepts will go unused.

   (a) Line 3 **C**

   (b) Line 5 **B**

   (c) Line 10 **E**

   (d) Line 12 **L**

   (e) Line 14 **A**

   (f) Line 15 **H**

   (g) Line 16 **K**

   (h) Line 26 **F**

   (i) Line 28 **J**

   (j) Line 29 **D**

A.	accessor
B.	aggregation
C.	constant
D.	delegation
E.	inheritance
F.	generics
G.	hibernation
H.	mutator
I.	polymorphism
J.	promotion
K.	overriding
L.	shadowing

The following code shows the API of two classes `Mammal` and `Dog`, and a class for the app `Zoo` which imports and uses `Mammal`.

```
public class Mammal
{
    public final int EYES = 2;
    public int mass;
    public Date birthday;

    public int eat(int x);     // Returns 2 * x.
}

public class Dog extends Mammal
{
    public long mass;

    public int  getAge();      // Returns the value of age.
    public void setAge(int x); // Changes the value of age.
    public int  eat(int x);    // Returns 7 * x.
    public int  eat(double x); // Returns (int)(7 * x).
}
⋮
import java.util.*;

public class Zoo
{
    public static void main(String[] args)
    {
        Set<Mammal> specimens = new HashSet<Mammal>();
        int x = 3;
        double y = x;
        int z = Integer.parseInt("416");
    }
}
```

2. Strings and Regular Expressions. [10pts]

(a) [2pts] Consider the following code.

```
String bookTitle = "The Count of Monty Python";
bookTitle.toLowerCase(); // Cast to lowercase.
System.out.println(nookTitle);
```

i. [1pt] What is immutability for `String`?

It refers to class with no mutator methods. So after an object of the class is constructed, its state cannot be changed.

ii. [1pt] What does the code above print?

The Count of Monty Python

(b) [2pts] A programmer has written the following, sentinel-based input loop. It is supposed to read input from the keyboard using a `Scanner` object named `input`, until the user enters the string `"done"` or `"DONE"`.

```
String userInput = input.next();
while (userInput != "done" && userInput != "DONE")
{
   do something
}
```

The program compiles and runs, but the loop body executes *even when* the user enters the sentinel string!

i. [1pt] Explain what is wrong with the programmer's code.

`userInput` is a different `String` object than the literals `"done"` and `"DONE"`.

ii. [1pt] Provide a fix (solution) for the problem.

Replace #2

```
while (!userInput.equals("done") && !userInput.equals("DONE"))
```

(c) `[4pts]` A programmer is using the fixed-length codes technique to convert York letter grades (`A`, `B`, `C`, `D`, `E`, `F`) to numeric GPA values (`8`, `6`, `4`, `2`, `1`, `0`).

```
final String LETTERS = "ABCDEF";
final String GPA     = "864210";
String grade = null;
Assume some code here that assigns a valid letter-grade to grade.
⋮
```

i. `[2pts]` Add one to three lines of Java code at line 5 onward to store in a variable of the appropriate type the correct numeric GPA value corresponding to `grade`'s value. E.g., if `grade = B`, then the GPA value is 6.

```
int pos = LETTERS.indexOf(grade);
String gradePoint = GPA.substring(pos, pos + 1);
```

ii. `[1pt]` What happens if `grade`'s value is a string of length one, but it does not correspond to one of the valid letter values?

It fails at runtime with an index-out-of-range exception in the `substring` call.

iii. `[1pt]` Suggest one way to deal with this problem. (Your answer does not need to include any Java code.)

Check with an `if...else...` whether -1 is returned by `indexOf`. Or use a `try...catch`.

(d) `[2pts]` Write a regular expression that will match a sequence of one or more lowercase English letters, and nothing else.

`[a-z]+` *or* `^[a-z]+$`

3. **Collections.** [10pts]

For each of the following,

- state what the program will print, *or*
- state that the program *fails* with a compile-time error, *and* clearly state *why*.

(a) [2pts]

```
Set<Integer> nums = new TreeSet<Integer>();
nums.add(5);
nums.add(10);
nums.add(2);
nums.add(5);

System.out.println("[" + nums.size() + "]");
for (Integer i : nums)
{
   System.out.println(i);
}
```

```
[3]
2
5
10
```

(b) [2pts]

```
Map<String, String> m = new HashMap<String, String>();
m.put("LOL", "laugh out loud");

for (String s : m.values())
{
   System.out.println(s);
}
```

```
laugh out loud
```

(c) [2pts]

```
Collection<RewardCard> cards = new ArrayList<RewardCard>();
cards.add(new CreditCard(1001, "Bob") );

System.out.println(cards.size());
```

It fails because `CreditCard` is not a sub-class of RewardCard*, the element type of the collection. (It is a super-class, but that does not help.)*

(d) [2pts]

```
List<Character> letters = new ArrayList<Character>();
letters.add('C');
letters.add('B');
letters.add('A');

for (int i = 0; i < 3; i++)
{
   System.out.println(letters.get(i));
}
```

```
C
B
A
```

(e) [2pts]

```
Set<String> s = new HashSet<String>();
s.add("A");
s.add("B");
s.add("C");

System.out.println(s.get(1));
```

It fails because `Set` has no method `get`. Retrieving by index does not make sense conceptually for sets.

4. **Inheritance.** [10pts]

Refer to the API for `Mammal` and `Dog` on page 3. For each code question, state what each prints, *or* that it fails at compile-time *or* run-time and *why*.

(a) [5pts]

i. [1pt]

```
Dog fred = new Dog();
System.out.println(fred.EYES);
```

2

ii. [1pt]

```
Dog fred = new Dog();
System.out.println(fred.eat(3.2));
```

22

iii. [1pt]

```
Mammal fred = new Dog();
System.out.println(fred.eat(3));
```

21

iv. [1pt]

```
Mammal fred = new Dog();
System.out.println(fred.eat(3.2));
```

Compile error because Mammal *has no method* ***`eat`*** *with signature* ***`double`****, which is checked for at early binding.*

v. [1pt]

```
Dog fred = new Dog();
System.out.println(fred.setAge(5).getAge());
```

Compile error because the return type of setAge *is* ***`void`****.*

(b) `[5pts]`

i. `[1pt]` What kind of relationship between classes does inheritance describe?

is a

ii. `[1pt]`

```
Dog fred = new Dog();
System.out.println(fred instanceof Mammal);
```

true

iii. `[1pt]` Say whether the following is correct, or explain why it is incorrect.

```
Map<Dog, int> pairs = new HashMap<Dog, int>();
Dog greg = new Dog();
pairs.add(greg, 7);
```

Compile-time error in line 1 because the map's value type must be a class.

iv. `[2pts]` What does substitutability mean in Java?

Wherever an object of a parent class is requested, an object of a child class will suffice.

5. **Exceptions & Validation.** `[10pts]`

Circle the letter corresponding to the one best answer for each.

(a) `[2pts]` A client must write code to deal with what kind of exceptions?

A. `Exception` and its subclasses.
B. `Error` and its subclasses.
C. `RuntimeException` and its subclasses.
$\boxed{\textbf{D.}}$ Checked exceptions.
E. None.

(b) `[2pts]` What kind of exceptions can usually be prevented from being thrown if the programmer performs input validation?

A. `Exception` and its subclasses.
B. `Error` and its subclasses.
$\boxed{\textbf{C.}}$ `RuntimeException` and its subclasses.
D. Checked exceptions.
E. None.

(c) `[2pts]` What output does the following code fragment produce?

```
for (int i = 3; i >= 0; i--)
{
   try
   {
      int quotient = 6 / i;
      System.out.print(quotient);
   }
   catch (ArithmeticException ex)
   {
      System.out.println(0);
   }
}
```

A. 236
$\boxed{\textbf{B.}}$ 2360
C. 0
D. *Compile-time error.*
E. *Run-time error.*

(d) `[2pts]` What output does the following code fragment produce?

```
try
{
    System.out.print("a");
    throw new RuntimeException("Arghh!");
}
catch(Exception err)
{
    System.out.print("b");
}
System.out.println("c");
```

A. ac
B. bc
$\boxed{\textbf{C.}}$ abc
D. *Compile-time error.*
E. *Run-time error.*

(e) `[2pts]` What output does the following code fragment produce?
(`NullPointerException` is a subclass of `RuntimeException`.)

```
try
{
   throw new NullPointerException();
}
catch (ArithmeticException ex)
{
   System.out.println("ArithmeticException!");
}
catch (IndexOutOfBoundsException ex)
{
   System.out.println("IndexOutOfBoundsException!");
}
catch (NoSuchElementException ex)
{
   System.out.println("NoSuchElementException!");
}
```

A. `ArithmeticException!`
B. `IndexOutOfBoundsException!`
C. `NoSuchElementException!`
D. *Compile-time error.*
$\boxed{\textbf{E.}}$ *Run-time error.*

Blank Page.

Relax. Breathe. Turn in your exam.