JAVA REVIEW QUESTIONS

Name : Hrudhay Reddy Garisa

OOPS Concepts Exercise Questions

Question 1

exercise: Mixed Messages

A short Java program is listed below. One block of the program is missing! Your challenge is to match the candidate block of code (on the left), with the output that you'd see if the block were inserted. Not all the lines of output will be used, and some of the lines of output might be used more than once. Draw lines connecting the candidate blocks of code with their matching command-line output.

The program:

```
class A {
  int ivar = 7;

  void m1() {
    System.out.print("A's m1, ");
  }
  void m2() {
    System.out.print("A's m2, ");
  }
  void m3() {
    System.out.print("A's m3, ");
  }
}

class B extends A {
  void m1() {
    System.out.print("B's m1, ");
  }
}

class C extends B {
  void m3() {
    System.out.print("C's m3, " + (ivar + 6));
  }
}

public class Mixed2 {
  public static void main(String[] args) {
    A a = new A();
    B b = new B();
    C c = new C();
    A a2 = new C();

    // Candidate code goes here (three lines)

  }
}
```

Code candidates:

```
b.m1();
c.m2();
a.m3();
```

```
c.m1();
c.m2();
c.m3();
```

```
a.m1();
b.m2();
c.m3();
```

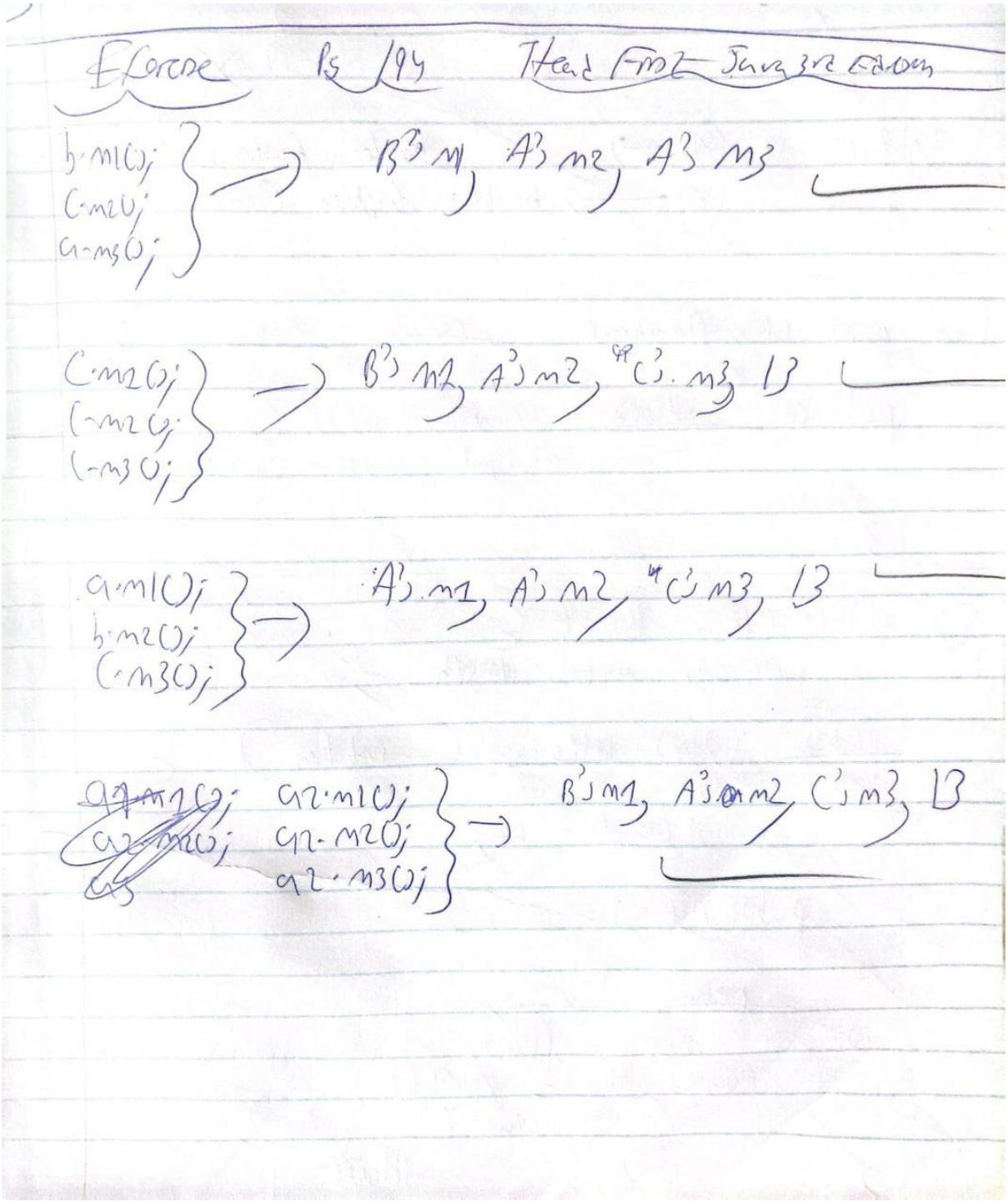
```
a2.m1();
a2.m2();
a2.m3();
```

Output:

```
A's m1, A's m2, C's m3, 6

B's m1, A's m2, A's m3,

A's m1, B's m2, A's m3,

B's m1, A's m2, C's m3, 13

B's m1, C's m2, A's m3,

B's m1, A's m2, C's m3, 6

A's m1, A's m2, C's m3, 13
```


Answers on page 197.

Answer to Question 1



Question 2

BE the Compiler

Which of the A-B pairs of methods listed on the right, if inserted into the classes on the left, would compile and produce the output shown? (The A method inserted into class Monster, the B method inserted into class Vampire.)

```
public class MonsterTestDrive {

  public static void main(String[] args) {
    Monster[] monsters = new Monster[3];
    monsters[0] = new Vampire();
    monsters[1] = new Dragon();
    monsters[2] = new Monster();
    for (int i = 0; i < monsters.length; i++) {
      monsters[i].frighten(i);
    }
  }
}

class Monster {
  A
}

class Vampire extends Monster {
  B
}

class Dragon extends Monster {
  boolean frighten(int degree) {
    System.out.println("breathe fire");
    return true;
  }
}
```

```
File  Edit  Window  Help  SaveYourself

% java MonsterTestDrive
a bite?
breathe fire
arrrgh
```

1

A
```
boolean frighten(int d) {
  System.out.println("arrrgh");
  return true;
}
```

B
```
boolean frighten(int x) {
  System.out.println("a bite?");
  return false;
}
```

2

A
```
boolean frighten(int x) {
  System.out.println("arrrgh");
  return true;
}
```

B
```
int frighten(int f) {
  System.out.println("a bite?");
  return 1;
}
```

3

A
```
boolean frighten(int x) {
  System.out.println("arrrgh");
  return false;
}
```

B
```
boolean scare(int x) {
  System.out.println("a bite?");
  return true;
}
```

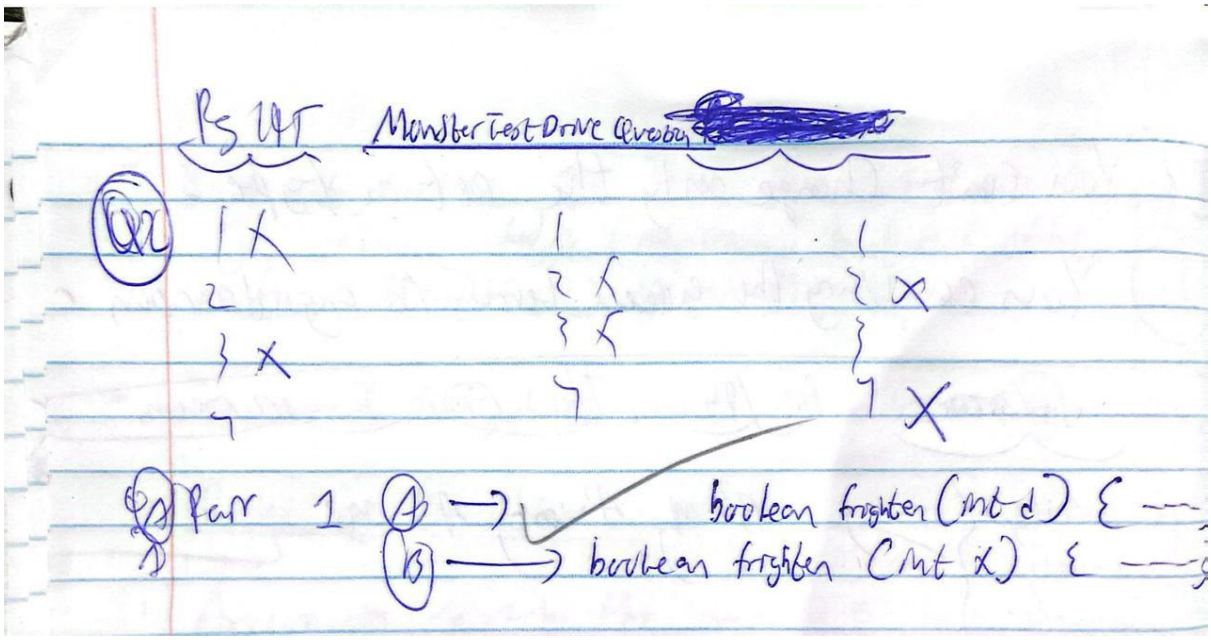
4

A
```
boolean frighten(int z) {
  System.out.println("arrrgh");
  return true;
}
```

B
```
boolean frighten(byte b) {
  System.out.println("a bite?");
  return true;
}
```

→ Answers on page 197.

Answer to Question 2



Question 3

puzzle: Pool Puzzle

Pool Puzzle

Your ***job*** is to take code snippets from the pool and place them into the blank lines in the code. You may use the same snippet more than once, and you might not need to use all the snippets. Your ***goal*** is to make a set of classes that will compile and run together as a program. Don't be fooled—this one's harder than it looks.

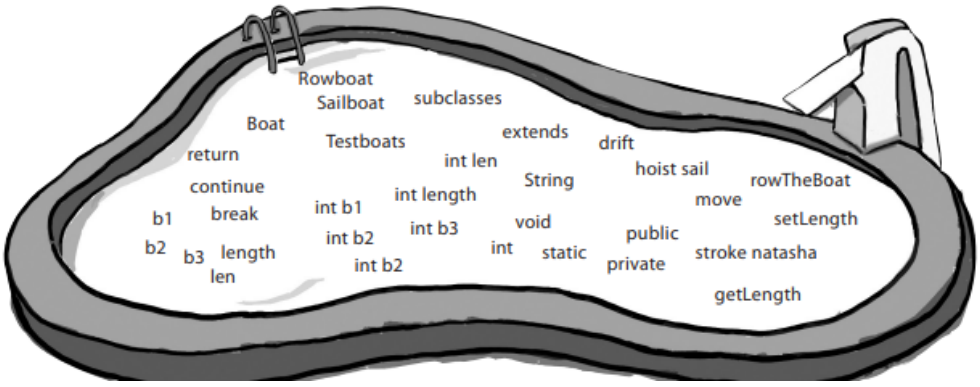
```
public class Rowboat _________ _________ {
   public ____________ rowTheBoat() {
      System.out.print("stroke natasha");
   }
}
```

```
public class _________ {
   private int ___________ ;
   _______ void _________ ( ______ ) {
      length = len;
   }
   public int getLength() {
      _________ __________ ;
   }
   public ____________ move() {
      System.out.print("____________");
   }
}
```

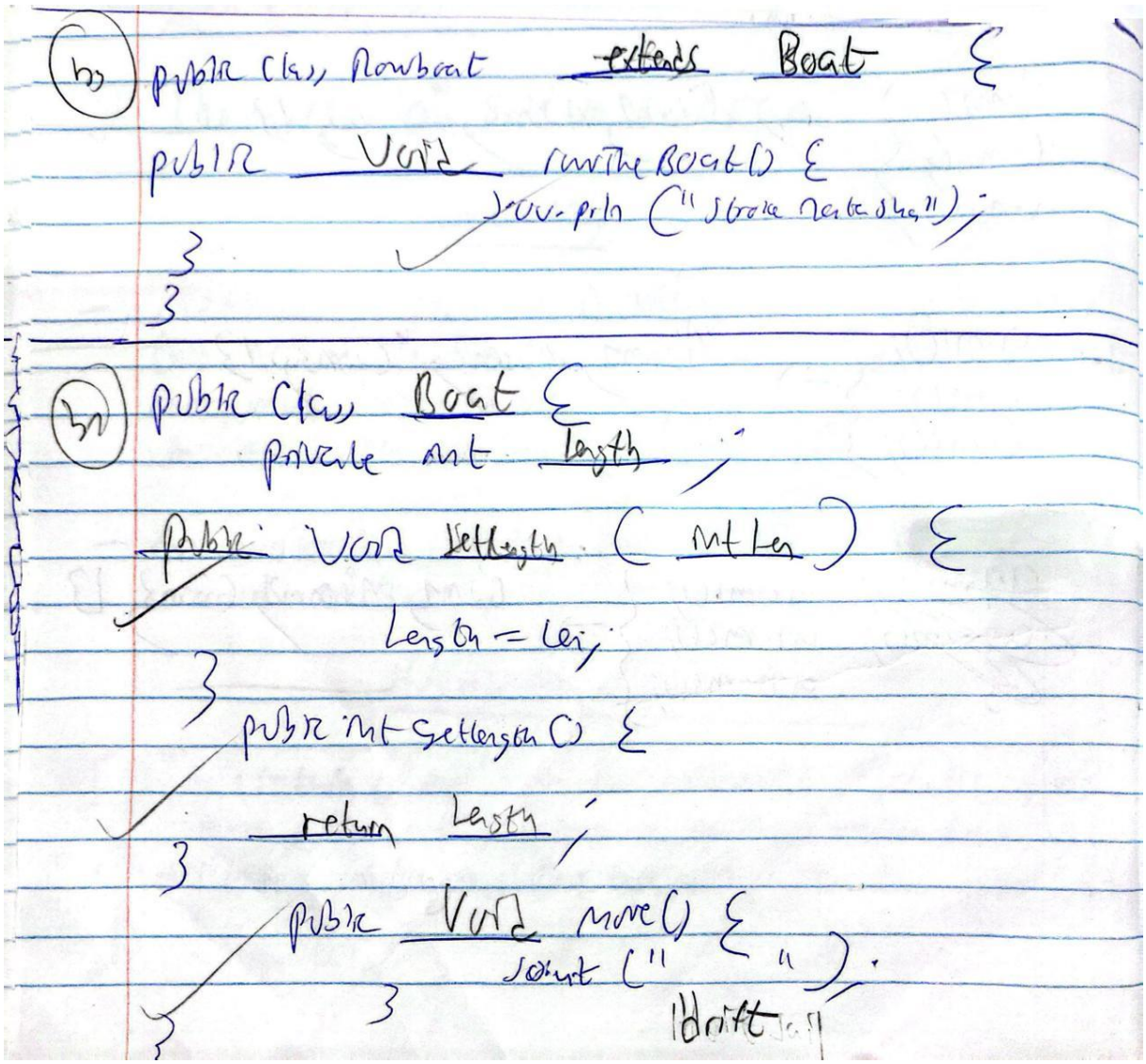
```
public class TestBoats {
   ______ ______ _______ main(String[] args){
      _________ b1 = new Boat();
      Sailboat b2 = new ___________();
      Rowboat _________ = new Rowboat();
      b2.setLength(32);
      b1.___________();
      b3.___________();
      _______.move();
   }
}
```

```
public class __________ _________ Boat {
   public _______ _________() {
      System.out.print("____________");
   }
}
```

OUTPUT: `drift drift hoist sail`



Answer to Question 3



```
public class TestBoat {

    public static void main (String [] args) {

        Boat b2 = new Boat();

        SailBoat b2 = new SailBoat();
        RowBoat b3 = new RowBoat();
        b2.setLength(32);
        b1.move();
        b3.move();
        b2.move();
    }

}
```

b2

```
public class SailBoat extends Boat {

    public void move() {

        System.out("hoist sail");

    }

}
```

Question 4

exercise: What's the Picture?

Here's your chance to demonstrate your artistic abilities. On the left you'll find sets of class and interface declarations. Your job is to draw the associated class diagrams on the right. We did the first one for you. Use a dashed line for "implements" and a solid line for "extends."

Given:

1.
```
public interface Foo { }
public class Bar implements Foo { }
```

2.
```
public interface Vinn { }
public abstract class Vout implements Vinn { }
```

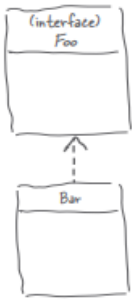
3.
```
public abstract class Muffie implements Whuffie { }
public class Fluffie extends Muffie { }
public interface Whuffie { }
```

4.
```
public class Zoop { }
public class Boop extends Zoop { }
public class Goop extends Boop { }
```

5.
```
public class Gamma extends Delta implements Epsilon { }
public interface Epsilon { }
public interface Beta { }
public class Alpha extends Gamma implements Beta { }
public class Delta { }
```

What's the Picture ?

1.

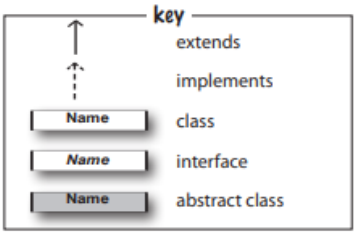


2.

3.

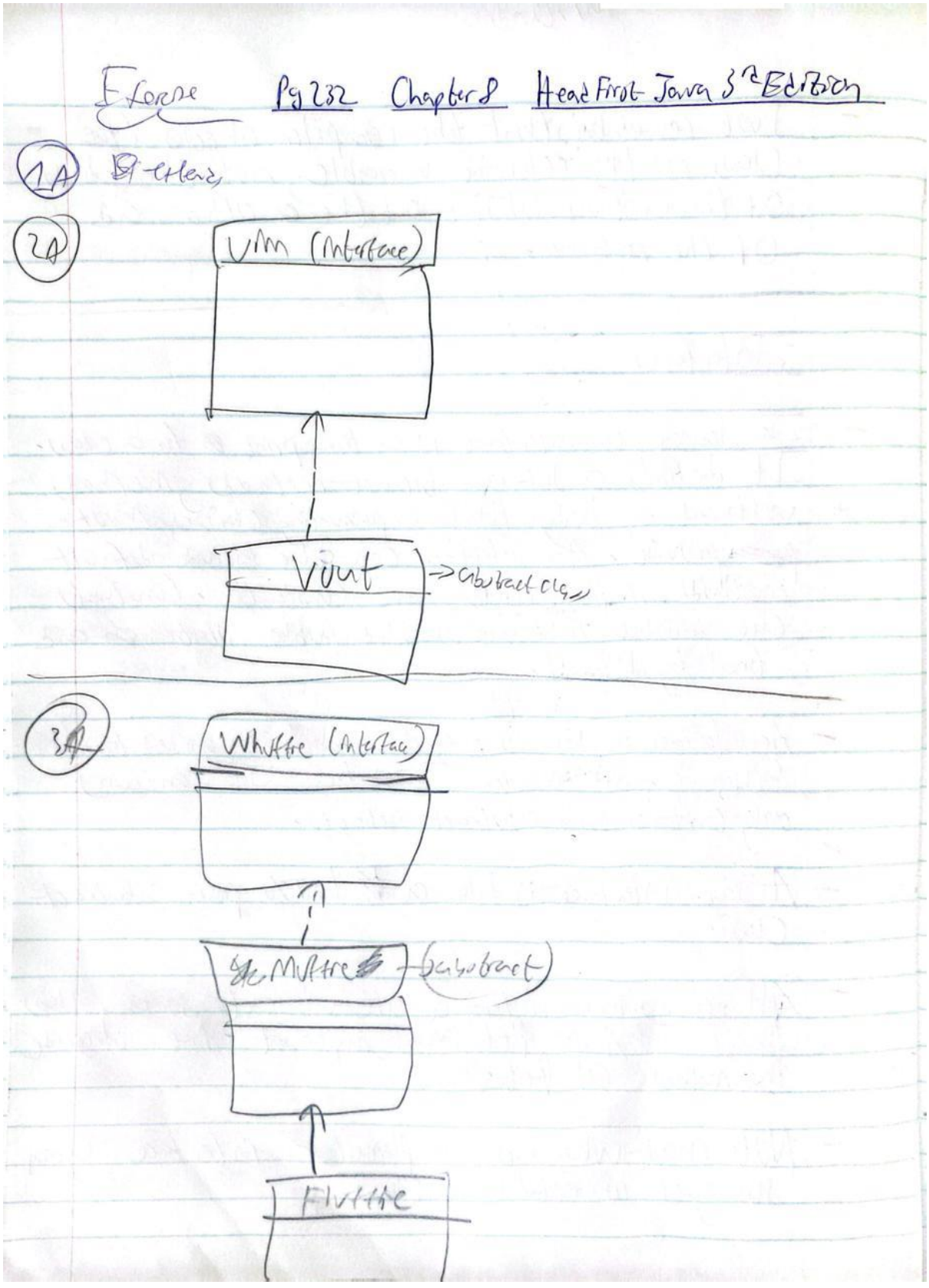
4.

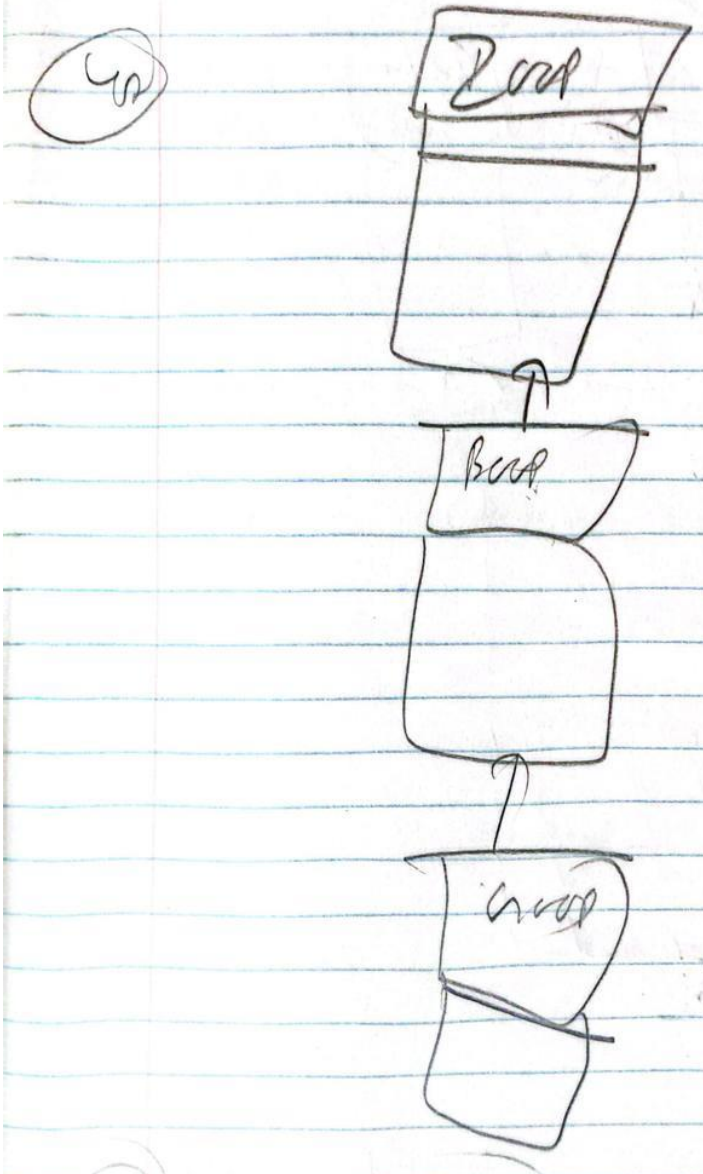
5.

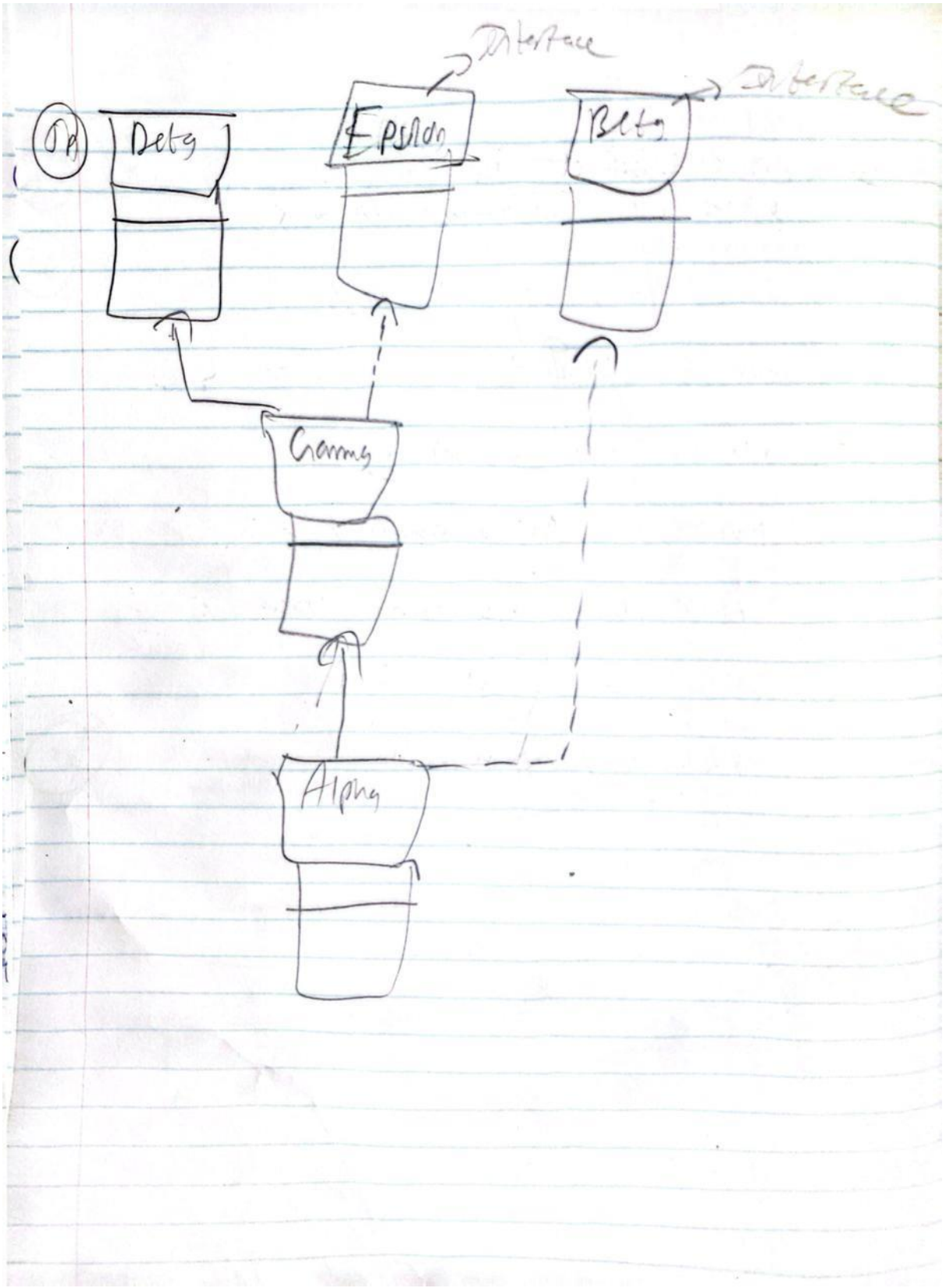


→ Answers on page 235.

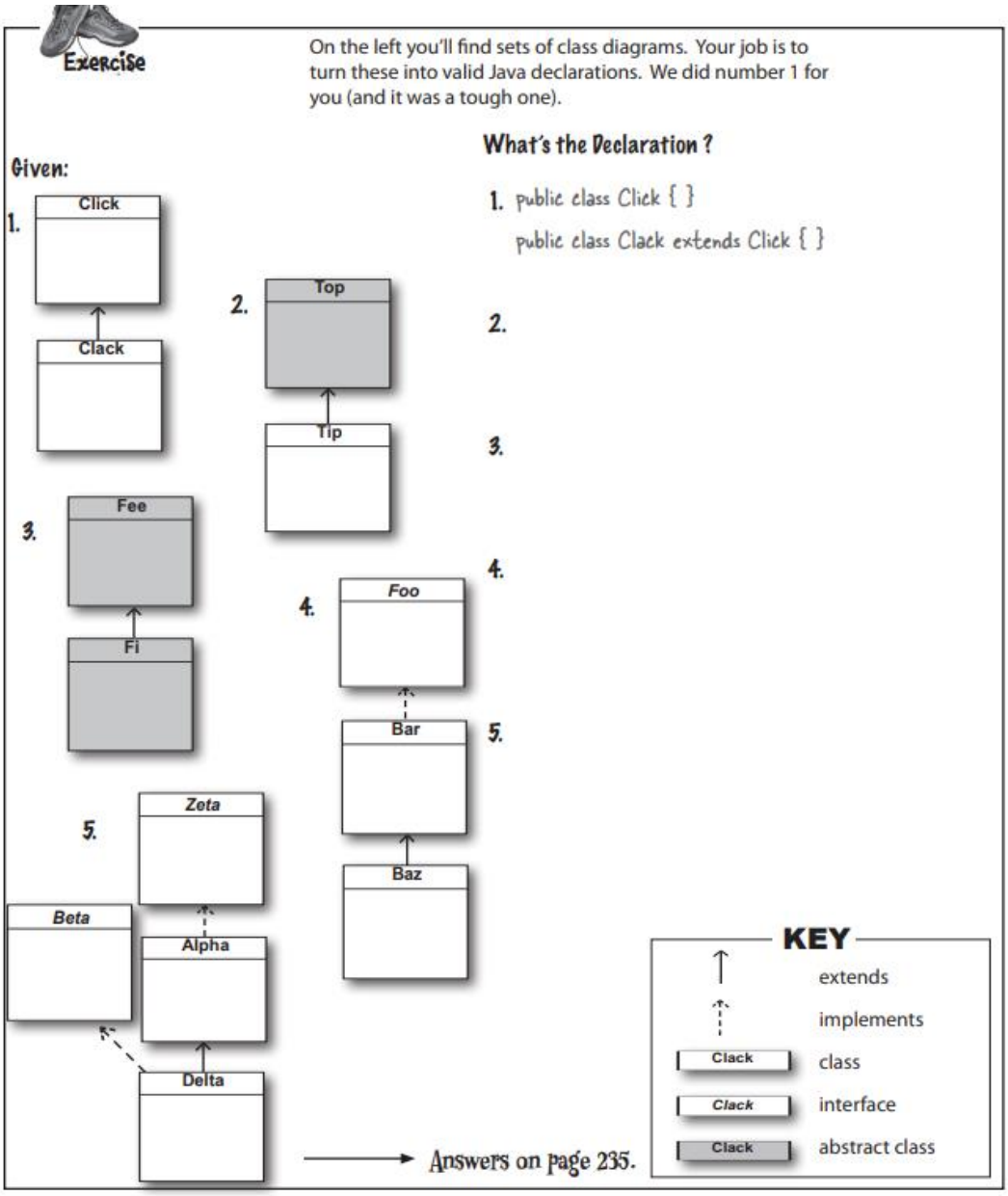
Answer to Question 4



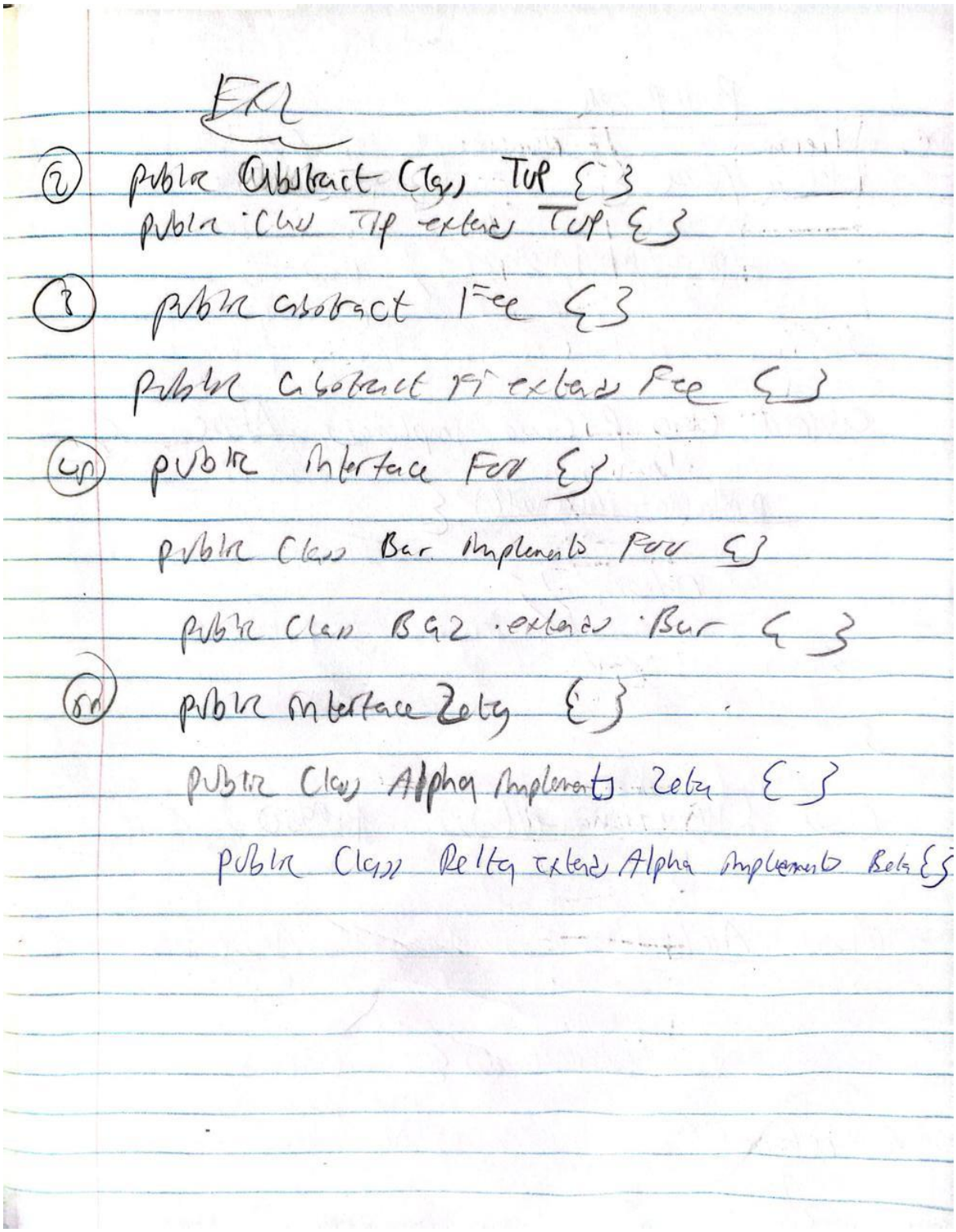



Interface
Interface
Delta
Epsilon
Beta
Gamma
Alpha

Question 5


Exercise
On the left you'll find sets of class diagrams. Your job is to turn these into valid Java declarations. We did number 1 for you (and it was a tough one).
What's the Declaration ?
Given:
1.
Click
Clack
1. public class Click { }
public class Clack extends Click { }
2.
Top
Tip
2.
3.
Fee
Fi
3.
4.
Foo
Bar
Baz
4.
5.
5.
Zeta
Beta
Alpha
Delta
KEY
extends
implements
Clack
class
Clack
interface
Clack
abstract class
Answers on page 235.

Answer to Question 5



Question 6

puzzle: Pool Puzzle

Your ***job*** is to take code snippets from the pool and place them into the blank lines in the code and output. You **may** use the same snippet more than once, and you won't need to use all the snippets. Your ***goal*** is to make a set of classes that will compile and run and produce the output listed.

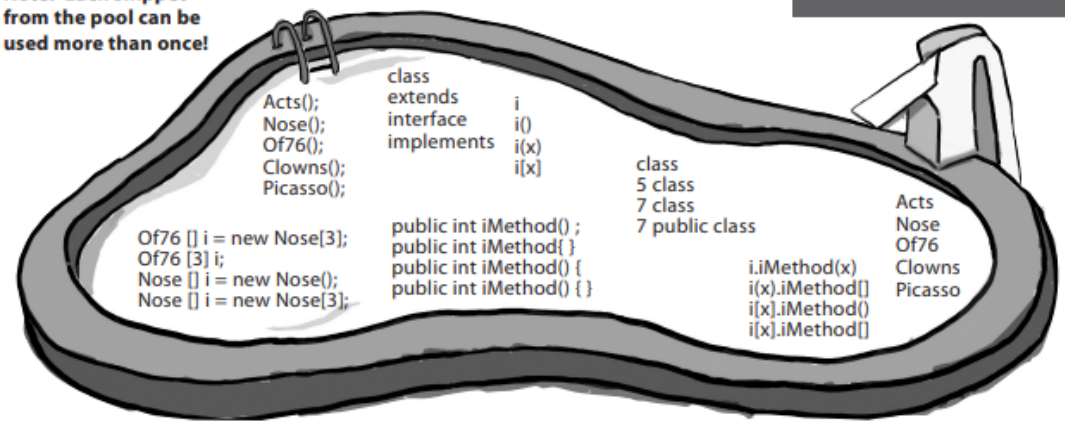
```
_____ Nose {
  _____
}

abstract class Picasso implements _____{
  _____
    return 7;
  }
}

class _____ _____ _____ { }

class _____ _____ _____ {
  _____
    return 5;
  }
}
```

```
public _____ _____ extends Clowns {

  public static void main(String[] args) {
    _____
    i[0] = new _____
    i[1] = new _____
    i[2] = new _____
    for (int x = 0; x < 3; x++) {
      System.out.println(_____
            + " " + _____.getClass());
    }
  }
}
```

Output

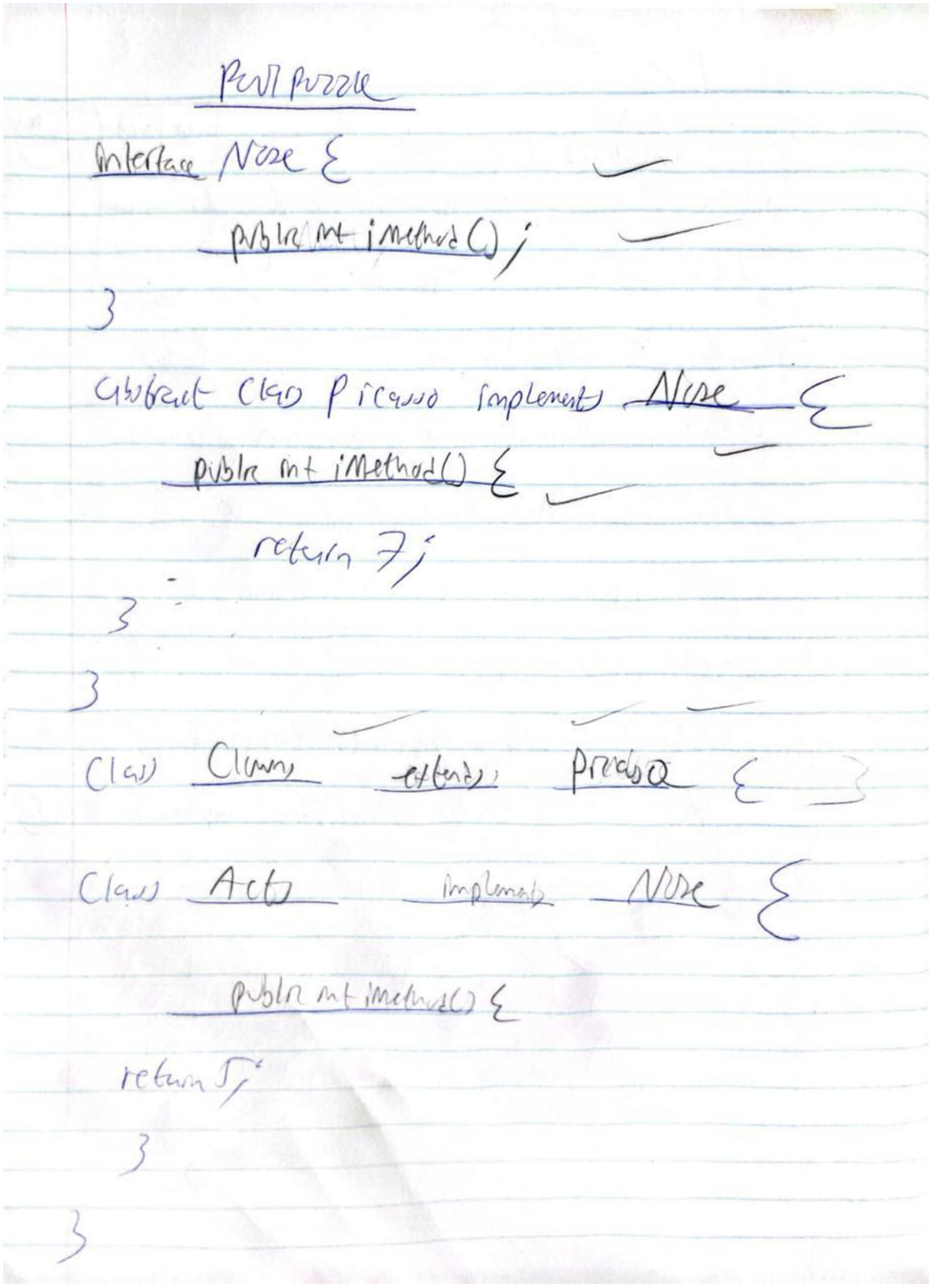
```
File Edit Window Help BeAfraid
%java _____
5 class Acts
7 class Clowns
_____Of76
```

Note: Each snippet from the pool can be used more than once!



Answers on page 236.

Answer to Question 6



```
Devil Puzzle

interface Nose {
    public int iMethod();
}

Abstract Class Picasso implements Nose {
    public int iMethod() {
        return 7;
    }
}

Class Clown extends Picasso { }

Class Actor implements Nose {
    public int iMethod() {
        return 5;
    }
}
```

public class Class76 extends Class1 {

public static void main(String[] args) {

Node[] i = new Node[3];

i[0] = new Abc();

i[1] = new Class1();

i[2] = new Class76();

for(int x=0; x<3; x++) {

System.out.println(i[x].iMethodA() + " " + i[x].getClass());

}

}

}

Output

Class Abc

Class Class1

Class Class76

Object Lifetime Review Questions

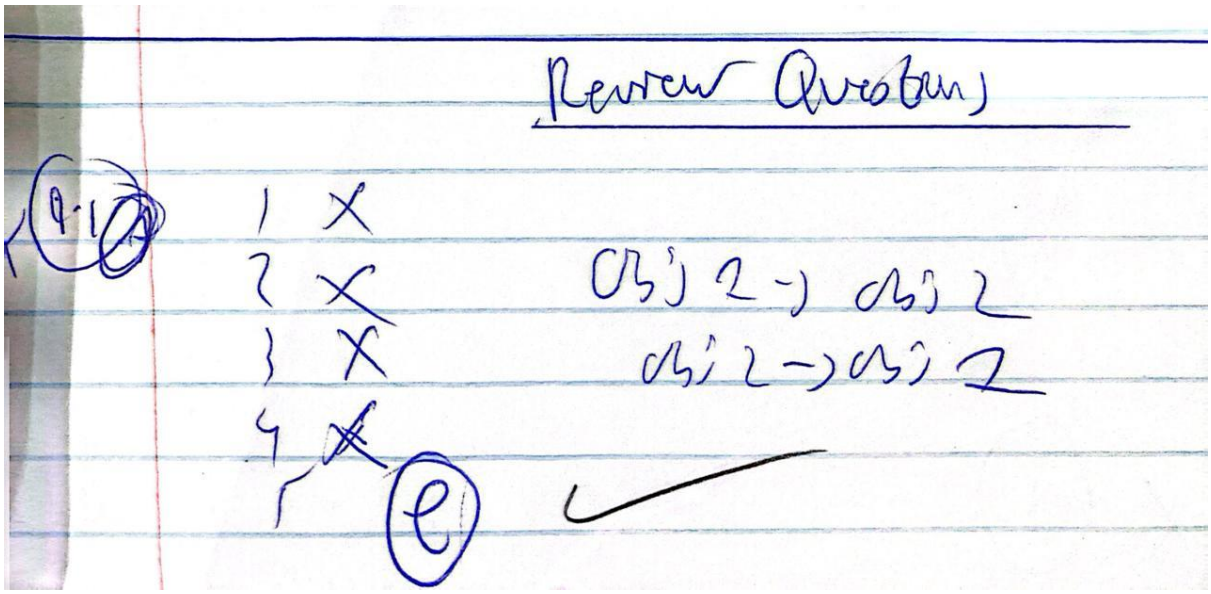
Question 1

9.1 Which statement is true?

Select the one correct answer.

(a) Objects can be explicitly destroyed using the keyword `delete`.
(b) An object will be garbage collected immediately after it becomes unreachable.
(c) If object `obj1` is accessible from object `obj2`, and object `obj2` is accessible from `obj1`, then `obj1` and `obj2` are not eligible for garbage collection.
(d) Once an object has become eligible for garbage collection, it will remain eligible until it is destroyed.
(e) If object `obj1` can access object `obj2` that is eligible for garbage collection, then `obj1` is also eligible for garbage collection.

Answer to Question 1



Question 2

9.2 Identify the location in the following program where the object, initially referenced with `arg1`, is eligible for garbage collection.

```
public class MyClass {
  public static void main(String[] args) {
    String msg;
    String pre = "This program was called with ";
    String post = " as first argument.";
    String arg1 = new String((args.length > 0) ? "'" + args[0] + "'" :
                             "<no argument>");
    msg = arg1;
    arg1 = null;              // (1)
    msg = pre + msg + post;   // (2)
    pre = null;               // (3)
    System.out.println(msg);
    msg = null;               // (4)
    post = null;              // (5)
    args = null;              // (6)
  }
}
```


Select the one correct answer.

(a) After the line labeled (1).
(b) After the line labeled (2).
(c) After the line labeled (3).
(d) After the line labeled (4).
(e) After the line labeled (5).
(f) After the line labeled (6).

Answer to Question 2

Question 3

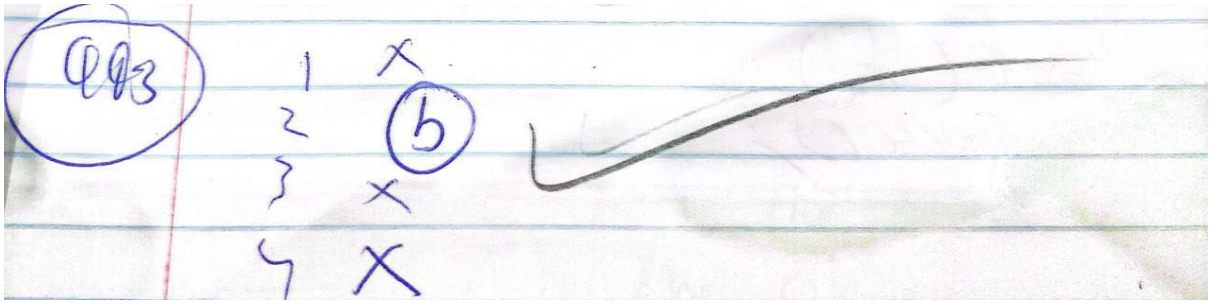
9.3 How many objects are eligible for garbage collection when control reaches (1)?

```
public class Eligible {
  public static void main(String[] args) {
    for (int i = 0; i < 5; i++) {
      Eligible obj = new Eligible();
      new Eligible();
    }
    System.gc();      // (1);
  }
}
```

Select the one correct answer.

(a) 0.
(b) 5.
(c) 10.
(d) Hard to say.

Answer to Question 3



Question 4

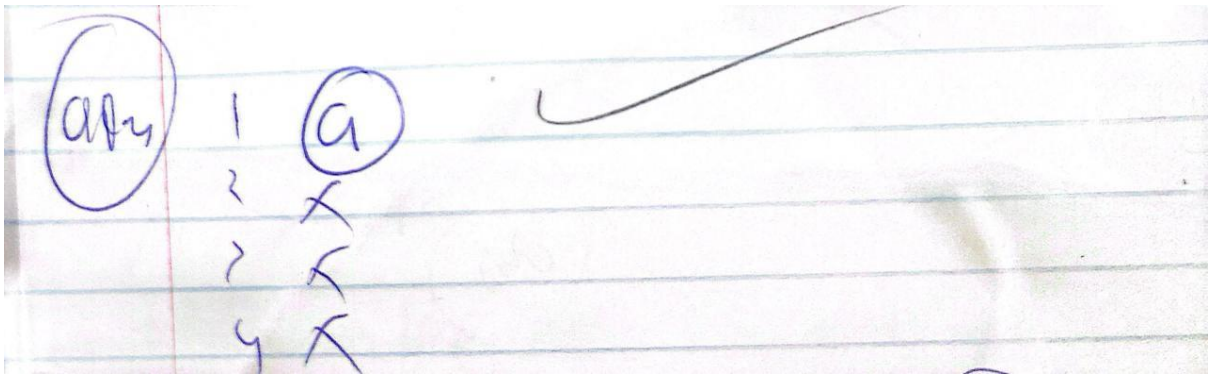
9.4 How many objects are eligible for garbage collection when control reaches (1)?

```
public class Link {
  private Link next;
  Link(Link next) { this.next = next; }
  public void finialize() { System.out.print("X"); }

  public static void main(String[] args) {
    Link p = null;
    for (int i = 0; i < 5; i++) {
      p = new Link(p);
    }
    System.gc();                    // (1);
  }
}
```

Select the one correct answer.

(a) 0
(b) 5
(c) 10
(d) Hard to say.

Answer to Question 4



Question 5

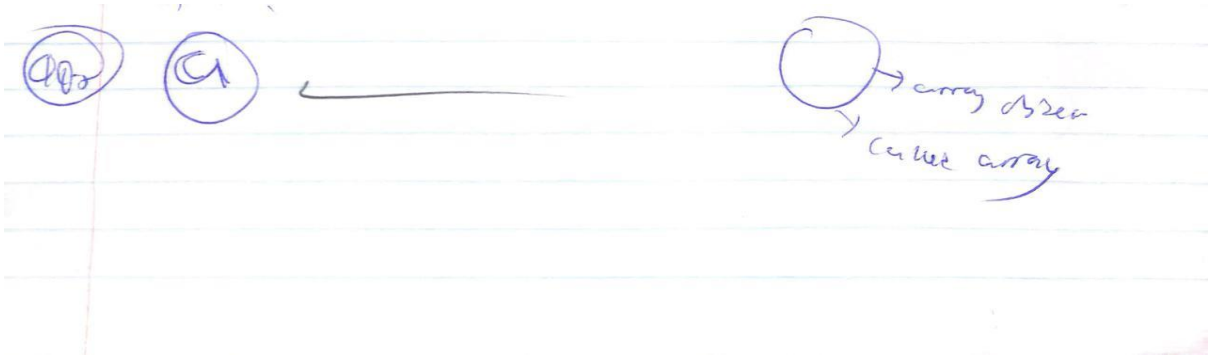
9.5 How many objects are eligible for garbage collection when control reaches (1)?

```
public class Elements {
  public static void main(String[] args) {
    int[] array = new int[4];
    for (int i = 0; i < 4; i++) {
      array[i] = i;
    }
    array[0] = array[1] = array[2] = array[3] = 0;
    System.gc();                              // (1);
  }
}
```

Select the one correct answer.

(a) 0
(b) 1
(c) 4
(d) Hard to say.

Answer to Question 5



Question 6

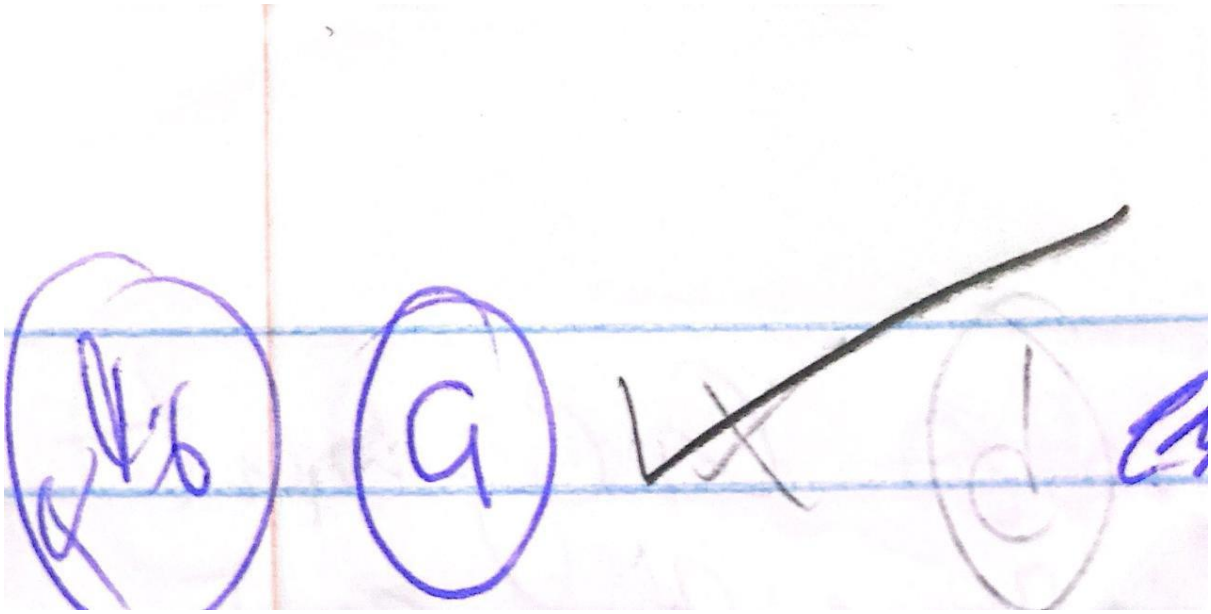
9.6 How many objects are reachable when control reaches (1)?

```
public class Nullify {

  private static void nullify(Integer[] array) { array = null; }

  public static void main(String[] args) {
    Integer[] array = new Integer[4];
    for (int i = 0; i < 4; i++) {
      array[i] = i;
    }
    nullify(array);
    System.gc();                          // (1);
  }
}
```

Select the one correct answer.

(a) 0
(b) 1
(c) 4
(d) 5
(e) Hard to say.

Answer to Question 6



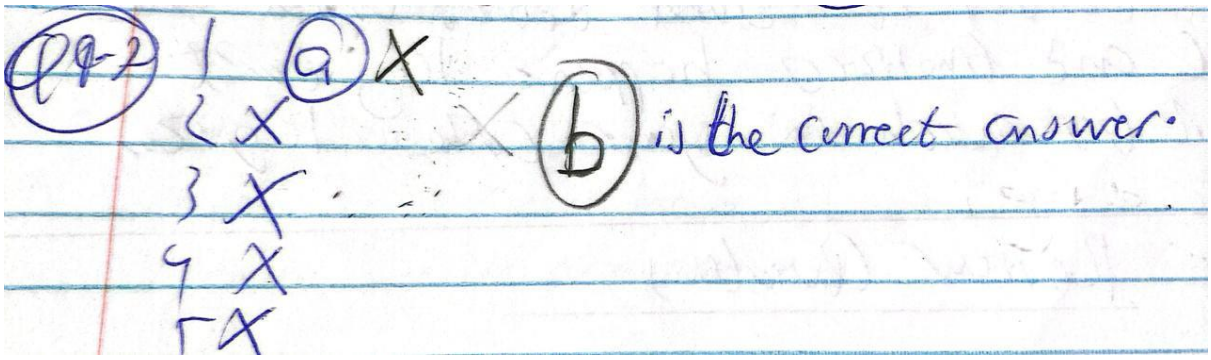
Question 7

9.7 Which statement is true?

Select the one correct answer.

(a) If an exception is thrown during the execution of the `finalize()` method of an eligible object, the exception is ignored and the object is destroyed.
(b) All objects have a `finalize()` method.
(c) Objects can be destroyed by explicitly calling the `finalize()` method.
(d) The `finalize()` method can be declared with any accessibility.
(e) The compiler will fail to compile code that defines an overriding `finalize()` method that does not explicitly call the overridden `finalize()` method from the superclass.

Answer to Question 7



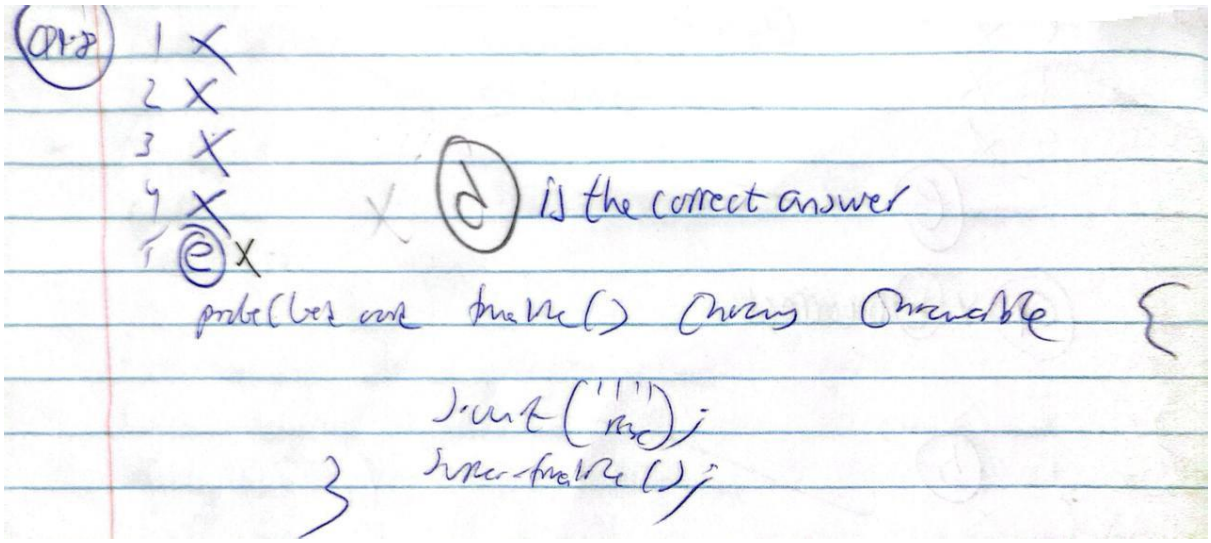
Question 8

9.8 Which statement is true?

Select the one correct answer.

(a) The compiler will fail to compile code that explicitly tries to call the `finalize()` method.
(b) The `finalize()` method must be declared with `protected` accessibility.
(c) An overriding `finalize()` method in any class can always throw checked exceptions.
(d) The `finalize()` method can be overloaded.
(e) The body of the `finalize()` method can only access other objects that are eligible for garbage collection.

Answer to Question 8



Question 9

9.9 Which statement describes guaranteed behavior of the garbage collection and finalization mechanisms?

Select the one correct answer.

(a) Objects will not be destroyed until they have no references to them.
(b) The `finalize()` method will never be called more than once on an object.
(c) An object eligible for garbage collection will eventually be destroyed by the garbage collector.
(d) If object `A` became eligible for garbage collection before object `B`, then object `A` will be destroyed before object `B`.
(e) An object, once eligible for garbage collection, can never become accessible by a live thread.

Question 10

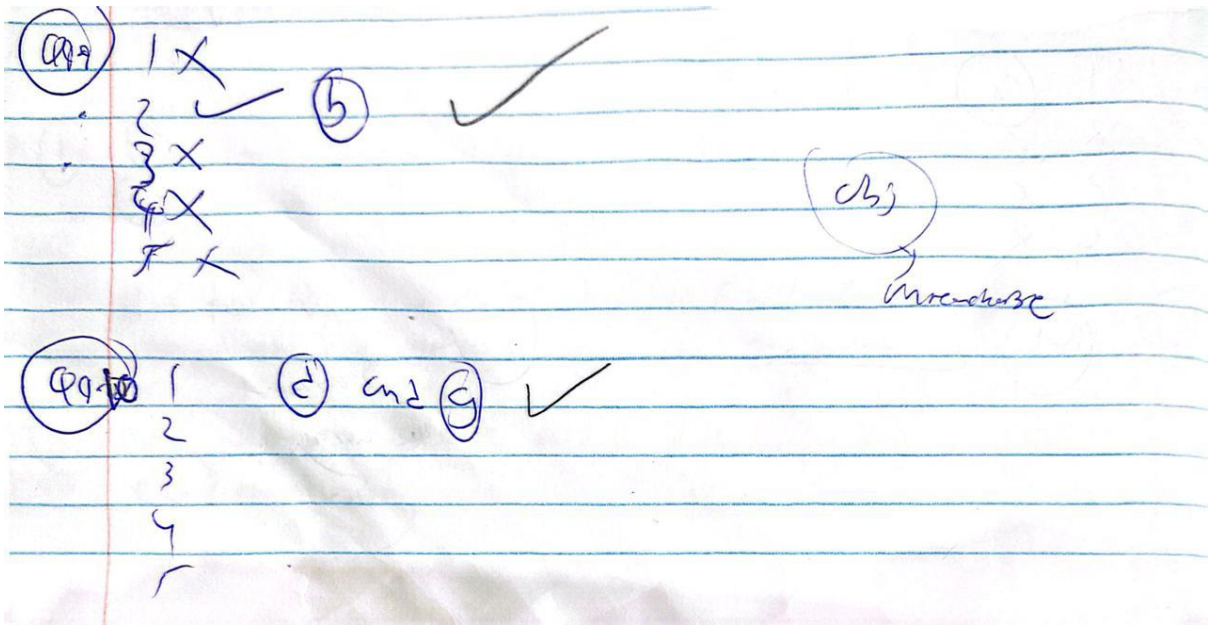
9.10 Which method headers will result in a correct implementation of a finalizer for the following class?

```
public class Curtain {
  // (1) INSERT METHOD HEADER HERE ...
  {
    System.out.println("Final curtain");
    super.finalize();
  }
}
```

Select the two correct answers.

(a) `void finalize() throws Throwable`
(b) `void finalize() throws Exception`
(c) `void finalize()`
(d) `protected void finalize() throws Throwable`
(e) `protected void finalize() throws Exception`
(f) `protected void finalize()`
(g) `public void finalize() throws Throwable`
(h) `public void finalize() throws Exception`
(i) `public void finalize()`
(j) `private void finalize() throws Throwable`
(k) `private void finalize() throws Exception`
(l) `private void finalize()`

Answer to Question 9 and 10



Question 11

9.11 Which scenario cannot definitely be the result of compiling and running the following program?

```
public class Grade {
  private char grade;
  Grade(char grade) { this.grade = grade; }

  public void finalize() throws Throwable {
    System.out.print(grade);
    super.finalize();
  }
  public static void main(String[] args) {
    new Grade('A'); new Grade('F');
    System.gc();
  }
}
```

Select the one correct answer.

(a) The program may print AF.
(b) The program may print FA.
(c) The program may print A.
(d) The program may print F.
(e) The program may print AFA.
(f) The program may not print anything.

Question 12

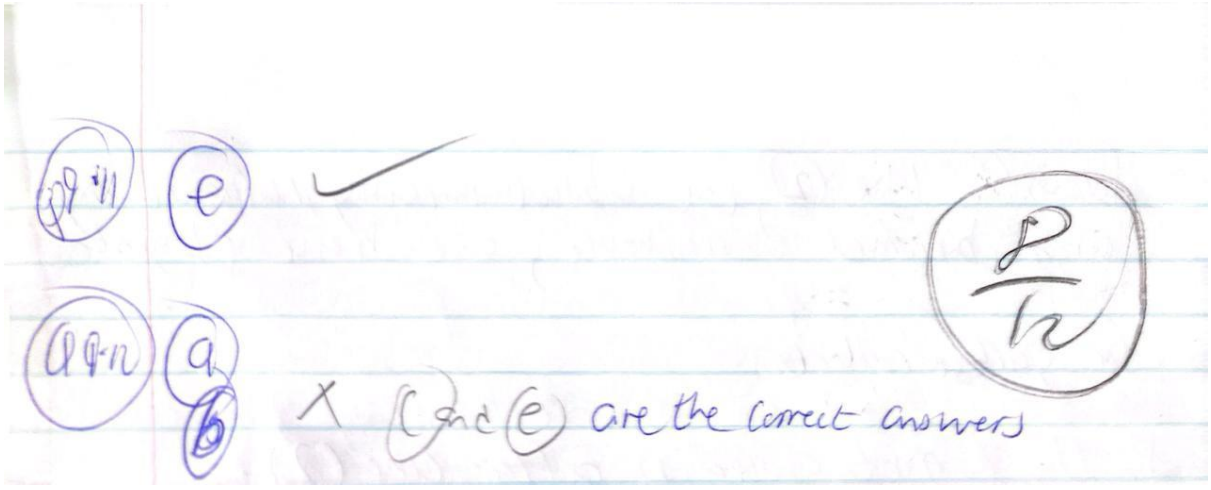
9.12 Which scenario can be the result of compiling and running the following program?

```
public class MyString {
  private String str;
  MyString(String str) { this.str = str; }

  public void finalize() throws Throwable {
    System.out.print(str);
    super.finalize();
  }

  public void concat(String str2) {
    this.str.concat(str2);
  }

  public static void main(String[] args) {
    new MyString("A").concat("B");
    System.gc();
  }
}
```

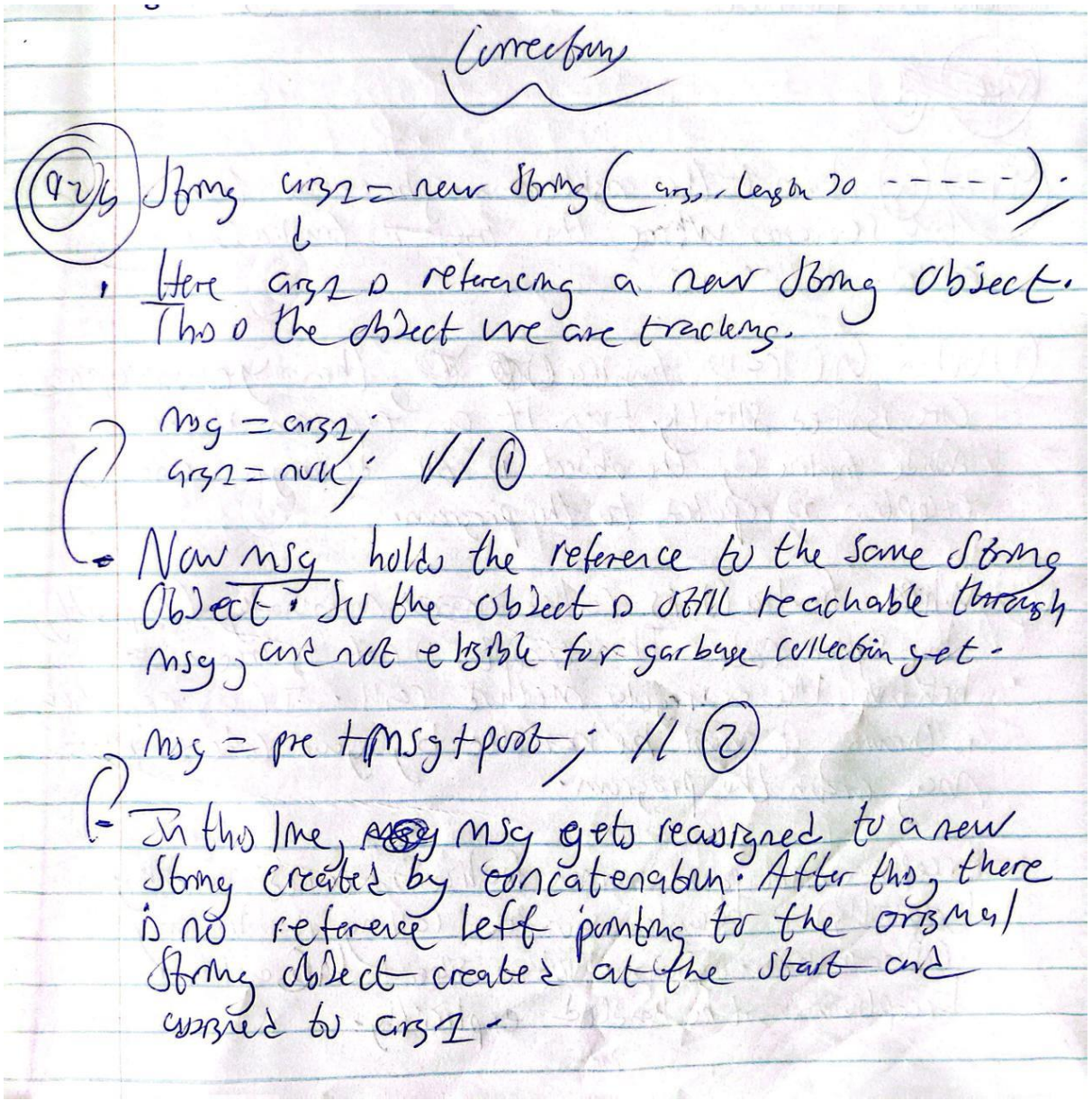
Select the two correct answers.

(a) The program may print AB.
(b) The program may print BA.
(c) The program may print A.
(d) The program may print B.
(e) The program may not print anything.

Answer to Question 11 and 12



Correction to Question 2



Correction

Q2/ String arg2 = new String(arg2, length 20 -----);

- Here arg2 is referencing a new String object. This is the object we are tracking.

msg = arg2;
arg2 = null; // ①

- Now msg holds the reference to the same String object. So the object is still reachable through msg, and not eligible for garbage collection yet.

msg = pre + msg + post; // ②

- In this line, msg gets reassigned to a new String created by concatenation. After this, there is no reference left pointing to the original String object created at the start and assigned to arg2.

- So after line ②, the object originally referenced by arg2 becomes unreachable, and hence eligible for garbage collection.

So the correct answer is after line ②.

Correction to Question 7

Q7 (A) is not the right answer, because there can be two scenarios where the ~~finalize~~ finalize() is being called.

(B) When GC calls finalize(), then yes, exceptions are ignored silently. Even if an exception is thrown inside finalize(), the object is still destroyed and no exception is reported to the program.

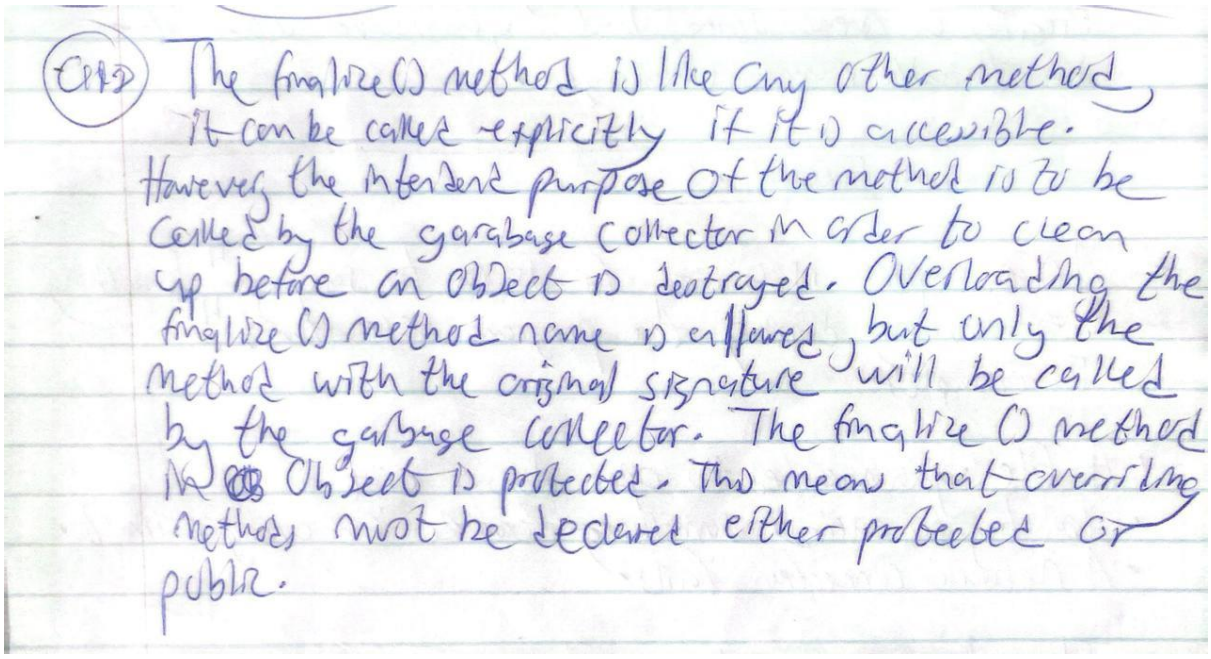
(C) When finalize() is called manually (explicitly) by your code. Then exception is not ignored. It behaves like a regular method call. If an exception is thrown, it must be handled by your code or it may crash the program.

Exception is not ignored when finalize() is called manually (explicitly) is because you're calling it like any other method in Java. Not a special GC action. Exceptions must be handled explicitly.

Still Q7

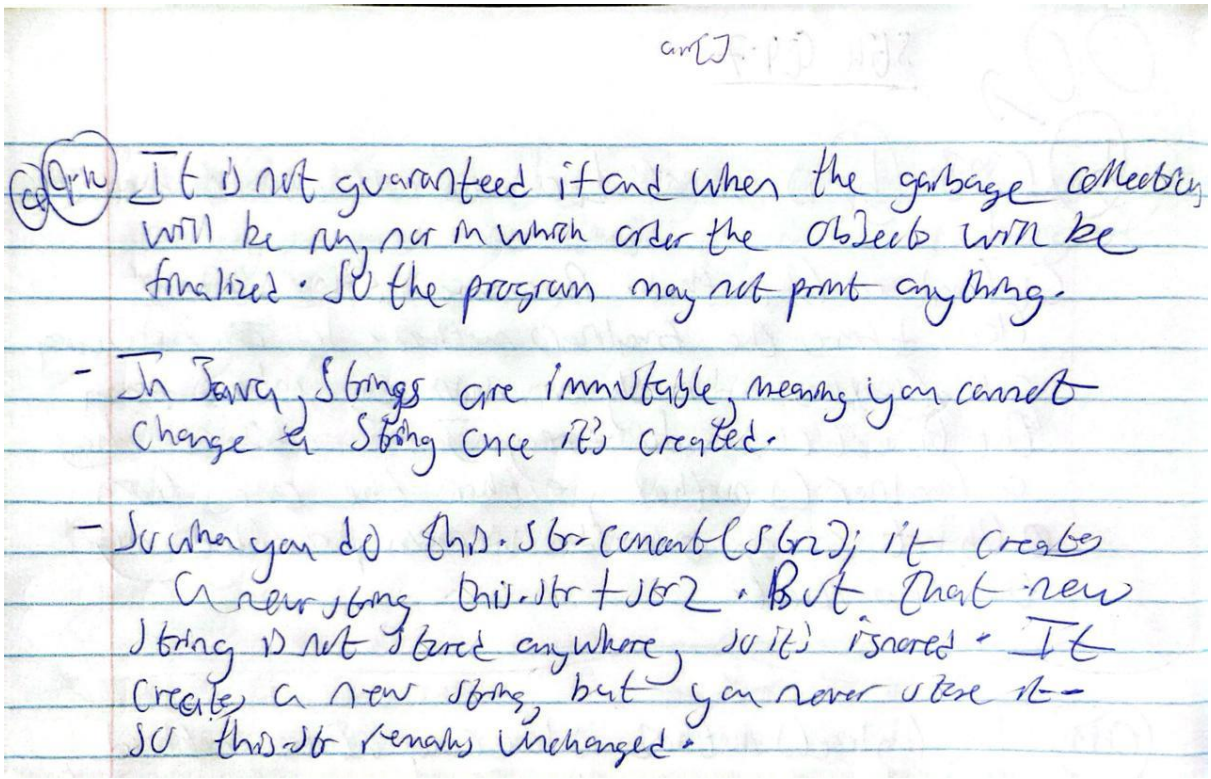
(b) Option (b) is correct because all objects have a finalize() method. Because the Object class defines the finalize() method. So in Java all classes directly or indirectly inherit from the Object class. So every Java object has a finalize() method, either the one from Object or an overridden version provided by the class itself.

Correction to Question 8



Q12 The finalize() method is like any other method, it can be called explicitly if it is accessible. However the intended purpose of the method is to be called by the garbage collector in order to clean up before an object is destroyed. Overloading the finalize() method name is allowed, but only the method with the original signature will be called by the garbage collector. The finalize() method in Object is protected. This means that overriding methods must be declared either protected or public.

Correction to Question 12



Q12 It is not guaranteed if and when the garbage collector will be run, nor in which order the objects will be finalized. So the program may not print anything.

- In Java, Strings are immutable, meaning you cannot change a String once it's created.
- So when you do this.str.concat(str2); it creates a new string this.str+str2. But that new String is not stored anywhere, so it's ignored. It creates a new string, but you never use it - so this.str remains unchanged.

Exception Handling Review Questions

Question 1

6.23 Which digits, and in what order, will be printed when the following program is run?

```
public class MyClass {
  public static void main(String[] args) {
    int k=0;
    try {
      int i = 5/k;
    } catch (ArithmeticException e) {
      System.out.println("1");
    } catch (RuntimeException e) {
      System.out.println("2");
      return;
    } catch (Exception e) {
      System.out.println("3");
    } finally {
      System.out.println("4");
    }
    System.out.println("5");
  }
}
```

Select the one correct answer.

(a) The program will only print 5.
(b) The program will only print 1 and 4, in that order.
(c) The program will only print 1, 2, and 4, in that order.
(d) The program will only print 1, 4, and 5, in that order.

(e) The program will only print 1, 2, 4, and 5, in that order.
(f) The program will only print 3 and 5, in that order.

Question 2

6.24 Given the following program, which statements are true?

```
public class Exceptions {
  public static void main(String[] args) {
    try {
      if (args.length == 0) return;
      System.out.println(args[0]);
    } finally {
      System.out.println("The end");
    }
  }
}
```

Select the two correct answers.

(a) If run with no arguments, the program will produce no output.
(b) If run with no arguments, the program will print `"The end"`.
(c) The program will throw an `ArrayIndexOutOfBoundsException`.
(d) If run with one argument, the program will simply print the given argument.
(e) If run with one argument, the program will print the given argument followed by `"The end"`.

Question 3

6.25 What will be the result of attempting to compile and run the following program?

```
public class MyClass {
  public static void main(String[] args) {
    RuntimeException re = null;
    throw re;
  }
}
```

Select the one correct answer.

(a) The code will fail to compile because the `main()` method does not declare that it throws `RuntimeException` in its declaration.
(b) The program will fail to compile because it cannot throw `re`.
(c) The program will compile without error and will throw `java.lang.RuntimeException` when run.
(d) The program will compile without error and will throw `java.lang.NullPointerException` when run.
(e) The program will compile without error and will run and terminate without any output.

Question 4

6.26 Which statements are true?

Select the two correct answers.

(a) If an exception is not caught in a method, the method will terminate and normal execution will resume.

(b) An overriding method must declare that it throws the same exception classes as the method it overrides.

(c) The `main()` method of a program can declare that it throws checked exceptions.

(d) A method declaring that it throws a certain exception class may throw instances of any subclass of that exception class.

(e) `finally` blocks are executed if, and only if, an exception gets thrown while inside the corresponding `try` block.

Question 5

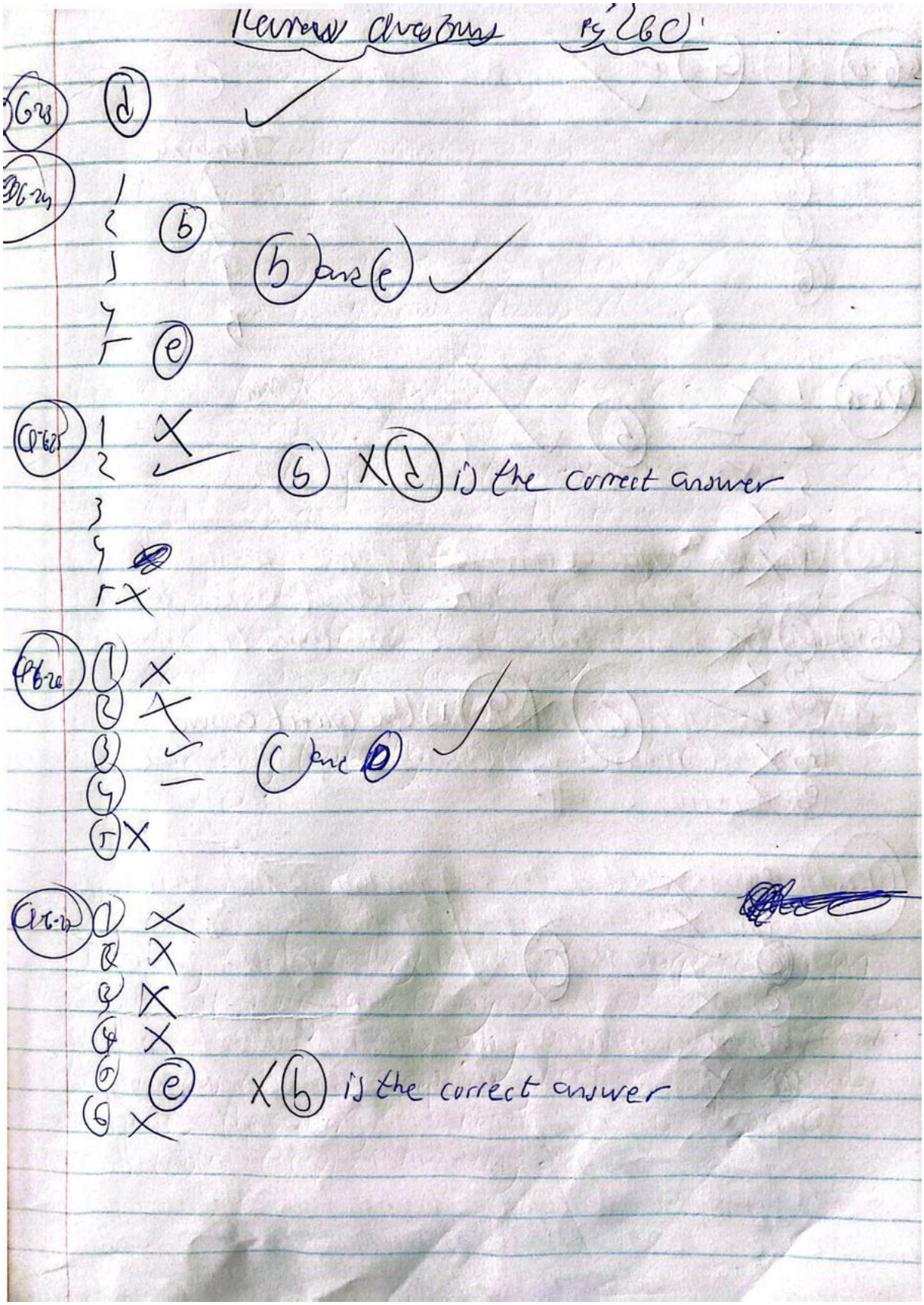
6.27 Which digits, and in what order, will be printed when the following program is compiled and run?

```
public class MyClass {
  public static void main(String[] args) {
    try {
      f();
    } catch (InterruptedException e) {
      System.out.println("1");
      throw new RuntimeException();
    } catch (RuntimeException e) {
      System.out.println("2");
      return;
    } catch (Exception e) {
      System.out.println("3");
    } finally {
      System.out.println("4");
    }
    System.out.println("5");
  }

  // InterruptedException is a direct subclass of Exception.
  static void f() throws InterruptedException {
    throw new InterruptedException("Time for lunch.");
  }
}
```

Select the one correct answer.

(a) The program will print 5.
(b) The program will print 1 and 4, in that order.
(c) The program will print 1, 2, and 4, in that order.
(d) The program will print 1, 4, and 5, in that order.
(e) The program will print 1, 2, 4, and 5, in that order.
(f) The program will print 3 and 5, in that order.

Answers to Question 1 , 2 , 3 , 4 and 5



Question 6

6.28 Which digits, and in what order, will be printed when the following program is run?

```
public class MyClass {
  public static void main(String[] args) throws InterruptedException {
    try {
      f();
      System.out.println("1");
    } finally {
      System.out.println("2");
    }
```



```
    System.out.println("3");
  }

  // InterruptedException is a direct subclass of Exception.
  static void f() throws InterruptedException {
    throw new InterruptedException("Time to go home.");
  }
}
```

Select the one correct answer.

(a) The program will print 2 and throw `InterruptedException`.
(b) The program will print 1 and 2, in that order.
(c) The program will print 1, 2, and 3, in that order.
(d) The program will print 2 and 3, in that order.
(e) The program will print 3 and 2, in that order.
(f) The program will print 1 and 3, in that order.

Question 7

6.29 What is wrong with the following code?

```
public class MyClass {
  public static void main(String[] args) throws A {
    try {
      f();
    } finally {
      System.out.println("Done.");
    } catch (A e) {
      throw e;
    }
  }

  public static void f() throws B {
    throw new B();
  }
}

class A extends Throwable {}

class B extends A {}
```

Select the one correct answer.

(a) The `main()` method must declare that it throws `B`.
(b) The `finally` block must follow the `catch` block in the `main()` method.
(c) The `catch` block in the `main()` method must declare that it catches `B` rather than `A`.
(d) A single `try` block cannot be followed by both a `finally` and a `catch` block.
(e) The declaration of class `A` is illegal.

Question 8

6.30 What is the minimal list of exception classes that the overriding method `f()` in the following code must declare in its `throws` clause before the code will compile correctly?

```
class A {
  // InterruptedException is a direct subclass of Exception.
  void f() throws ArithmeticException, InterruptedException {
```



```
    div(5, 5);
  }

  int div(int i, int j) throws ArithmeticException {
    return i/j;
  }
}

public class MyClass extends A {
  void f() /* throws [...list of exceptions...] */ {
    try {
      div(5, 0);
    } catch (ArithmeticException e) {
      return;
    }
    throw new RuntimeException("ArithmeticException was expected.");
  }
}
```

Select the one correct answer.

(a) Does not need to specify any exceptions.
(b) Needs to specify that it throws `ArithmeticException`.
(c) Needs to specify that it throws `InterruptedException`.
(d) Needs to specify that it throws `RuntimeException`.
(e) Needs to specify that it throws both `ArithmeticException` and `InterruptedException`.

Question 9

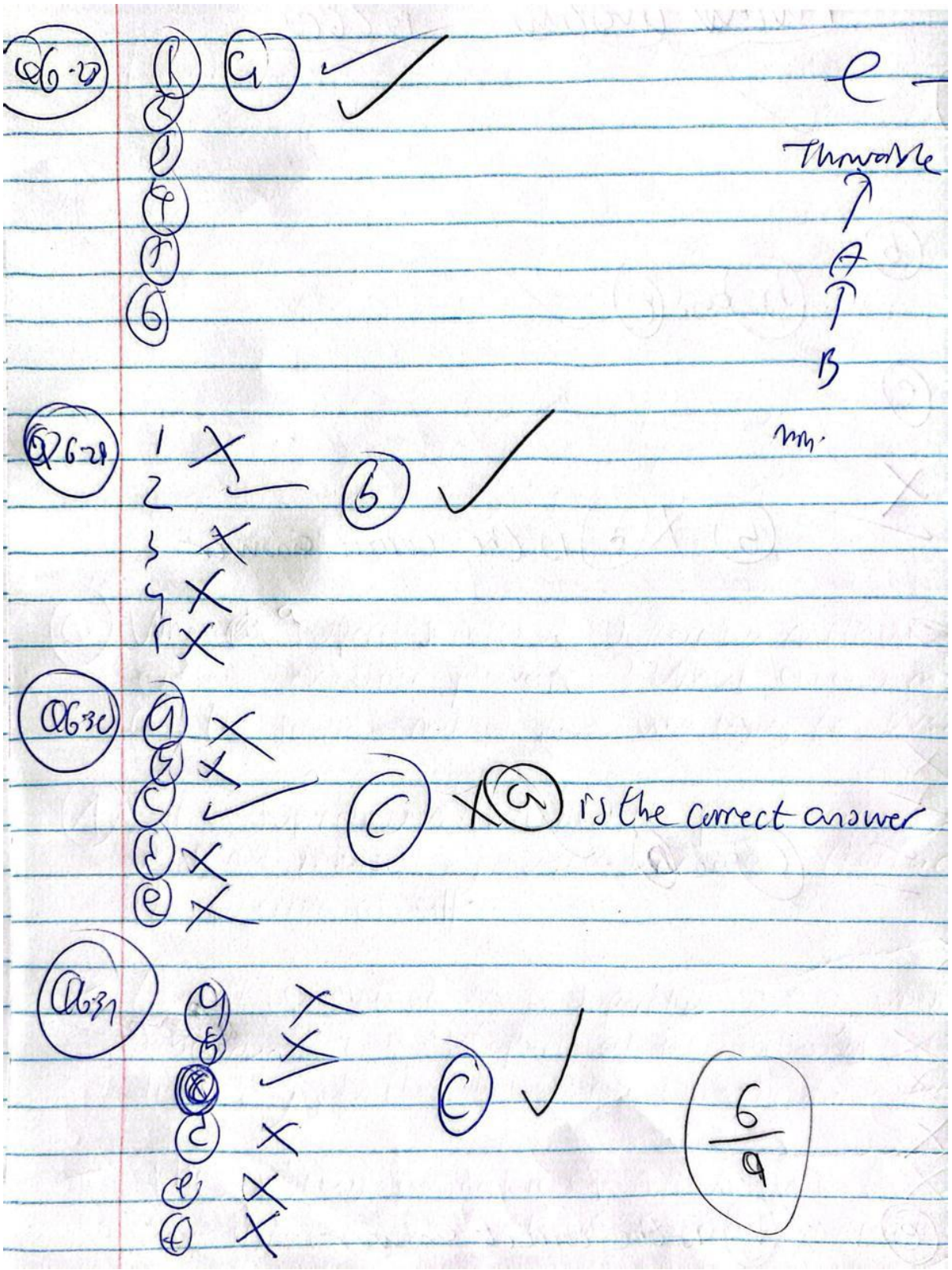
6.31 What, if anything, would cause the following code not to compile?

```
class A {
  void f() throws ArithmeticException {
    //...
  }
}

public class MyClass extends A {
  public static void main(String[] args) {
    A obj = new MyClass();

    try {
      obj.f();
    } catch (ArithmeticException e) {
      return;
    } catch (Exception e) {
      System.out.println(e);
      throw new RuntimeException("Something wrong here");
    }
  }

  // InterruptedException is a direct subclass of Exception.
  void f() throws InterruptedException {
    //...
  }
}
```


Select the one correct answer.

(a) The `main()` method must declare that it throws `RuntimeException`.
(b) The overriding `f()` method in `MyClass` must declare that it throws `ArithmeticException`, since the `f()` method in class `A` declares that it does.
(c) The overriding `f()` method in `MyClass` is not allowed to throw `InterruptedException`, since the `f()` method in class `A` does not throw this exception.
(d) The compiler will complain that the `catch(ArithmeticException)` block shadows the `catch(Exception)` block.
(e) You cannot throw exceptions from a `catch` block.
(f) Nothing is wrong with the code, it will compile without errors.

Answers to Question 6 , 7 , 8 and 9



Corrections for Question 3 , 5 and 8

Correction — R. From Review Questions

(Q6.24) The program will compile without error but will throw a NullPointerException when run. The throw statement can only throw Throwable objects. A NullPointerException will be thrown if the expression of the throw statement results in a null reference.

(d)

(Q6-27) (b) The program will print 1 and 4, in that order. An InterruptedException is handled in the first catch block. Inside this block a new RuntimeException is thrown. This exception was not thrown inside the try block and will not be handled by the catch block(s), but will be sent to the caller of the main () method. Before this happens, the finally block is executed. The code to print 5 is never reached, since the RuntimeException remains uncaught after the execution of the finally block.

(Q6.30) (A) Overriding methods can specify all, none, or a subset of the checked exceptions the overridden method declares in its throws clause. The InterruptedException is the only checked exception specified in the throws clause of the overridden method. The overriding method f() does not need to specify the InterruptedException from the throws clause of the overridden method, because the exception is not thrown there.