# Intermediate Programming Practice Exam 1

1. (2 points) What is your name?

For questions 2 through 6, please consider the following Java class:

```
class Student
{
        String name;
        int id;
        int points;
}
```

2. (5 points) In the space to the right, write a Java class named “Roster” that collects Student objects. (A roster is a list of students who are taking a particular course.) Give your Roster class a method named “add” that enables the user to add a Student object to the roster. (Please read questions 3 through 6 before you begin.)

3. (5 points) Add a member variable to your Roster class that holds the name of the course. Initialize this member variable with a value passed to the constructor.

4. (5 points) Add a method named “size”, which returns the number of Student objects currently in the Roster.

5. (5 points) Add a method named “getStudent”, which returns the Student object with a specified ID number. If no student in the roster has the specified ID, return null.

6. (5 points) Add a method named “getPoints”, which returns the number of points associated with the Student who has a specified ID number. If no student in the roster has the specified ID number, throw an informative RuntimeException (not a NullPointerException).

7. (5 points) Below, write a one-line Java command to instantiate a Roster object.

Roster rosty = new Roster(“Paradigms”);

```
class Roster
{
   LinkedList<Student> studs;
   String courseName;

   Roster(String cn)
   {
        courseName = cn;
        studs = new
   LinkedList<Student>();
   }

   void add(Student stu)
   {
        studs.add(stu);
   }

   int size()
   {
        return studs.size();
   }

   Student getStudent(int id)
   {
        Iterator<Student> it =
   studs.iterator();
        while(it.hasNext())
        {
            Student s = it.next();
            if(s.id == id)
                  return s;
        }
        return null;
   }

   int getPoints(int id)
   {
      Student stu = getStudent(id);
      if(stu == null)
        throw new
   RuntimeException(“uh oh”);
      return stu.points;
   }

}
```

8. (5 points) In this Java line of code,

```
    SomeClass sc;
```

the variable sc will be automatically set to

- 0
- new SomeClass()
- [✓ null]
- void
- whatever is in memory at the time
- class SomeClass

9. (5 points) Suppose you see a message like:

```
Exception in thread "main"
java.lang.NullPointerException
        at MyClass.<init>(MyClass.java:14)
        at MyClass.main(MyClass.java:27)
```

Which just occurred? (Circle one.)

- a compiler error
- [✓ a runtime error]

10. (5 points) How many objects of type “Yo” will the following code instantiate?

```
class Yo {
    static Yo mama = new Yo();
    Yo dawg;

    Yo() {
            dawg = null;
    }

    void whoa(Yo that) {
            if(this == that)
                System.out.println(“yo”);
    }

    public static void main(String[] args) {
            Yo ho;
            Yo yo;
            yo = new Yo();
            yo.whoa(ho);
            new Yo();//rk City
    }
}
```

3

11. (5 points) Static methods

- are called only once.
- are only instantiated once.
- can be compiled only once.
- can only be called from other static methods.
- [✓ have no implicit “this” parameter.]
- must be public.

12. (5 points) In Java, if you pass a variable of type “int” as a parameter to some function, it

- Implicitly calls the integer constructor.
- Passes the value in a member variable.
- Passes the reference in a member variable.
- Puts the memory location of the variable on the stack.
- [✓ Puts the value on the stack.]
- Puts the memory location of the variable on the heap.
- Puts the value on the heap.

13. (6 points) Look closely at this code. Especially, note the use of local variables in the constructor:

```
class Game {
   Avatar av;
   Background bg;
   Game() {
         av = new Avatar();
         bg = null;
   }

   void onClick(int x, int y) {
         if(x < 0 || y < 0)
            throw new
             RuntimeException(“!”);
         int i;
         av.teleport(x, y);
         if(bg == null)
              System.out.println(i);
   }
}
```

When “onClick” is called, a NullPointerException will be thrown. How would you fix it? (Circle one)

- Initialize bg with “new Background()”.
- [✓ Remove “`Avatar `” and “`Background `” from the constructor.]
- Pass valid values for x and y.
- Initialize “i” with “int i = 0;”.
- change the “==” to “!=”.
- Pass an Avatar object and a Background object as parameters to the constructor.
- Declare Avatar to be “static”.
- Declare Background to be “static”.

14. (5 points) In Java, if two variables reference the same object,

- a runtime exception will be thrown.
- the Java compiler will report an error.
- Java will make a deep copy of the first one.
- the garbage collector will remove one of the variables.
- this is not a problem, unless the programmer intended something else.
- a new object will be allocated for the second one.

15. (5 points) Suppose a class named Alpha extends a class named Beta. Which of these lines will cause a compile error?

- Alpha a = new Beta();
- Beta b = new Alpha();

16. (5 points) ConcurrentModificationException is thrown when

- You modify a variable that already references an object.
- Code is modified while you are debugging.
- Generic types are used improperly.
- An iterator detects that the contents of its collection have changed.
- You fail to use the “-g” flag when you build your code.
- You modify an object to which multiple variables refer.

17. (5 points) In Java, when a primitive type is passed as a parameter,

- its value is copied on the stack.
- its memory location is copied on the stack.
- a new object is allocated on the heap.
- the memory location of the object enclosing the primitive type is discovered by the garbage collector and is connected with the receiving function.

18. (6 points) Which two of the following statements are true?

- Abstract methods have no body.
- Abstract methods are not declared.
- Abstract classes cannot be extended.
- Abstract classes cannot be instantiated.
- Abstract classes cannot be compiled.

19. (6 points) Consider this nonsensical code:

```
class Fetuccini extends Object     // 1
{
        static Chicken c = new Chicken();
        AlfredoSauce a;               // 2

        void slurp(BreadStick b)
        {
              if(this.a == null)      // 3
                   this.b = null;      // 4
              else
                   c = null;          // 5
        }

        static AlfredoSauce swallow()// 6
        {
               if(c == null)          // 7
                       return this.a;   // 8
               else
                       return null;
        }
}
```

Of the eight commented lines, which two contain errors that the compiler will catch?

20. (5 points) Polymorphism happens when (circle the one best answer):

- The garbage collector cannot determine the type of an object.
- A type is specified to constrain a generic collection.
- An exception that was thrown unwinds the stack.
- An class that extends an abstract class is instantiated.
- The method that is called depends on the type of the “this” object.
- An error is handled at runtime instead of compile time.
- “this” is “null”.