

Example Questions for Practice

ECE373
Dr. Marefat

Multiple Choice (40 points, 5 each)

1. The default constructor that Java provides

- A. sets all the class's numeric fields to 0, boolean values to false, and reference variables to null.
- B. sets all the class's numeric fields to 0, boolean values to true, and reference variables to empty strings.
- C. sets all the class's numeric fields to 0, boolean values to true, and reference variables to null.
- D. sets all the class's numeric fields to 0, boolean values to false, and reference variables to empty strings.

2. If the following were all declared in the same class, *Rect*, which would be considered the no-arg constructor?

- (1) `public Rect(int len, int width){...}`
- (2) `public Rect(double len, double width){...}`
- (3) `public Rect(){...}`
- (4) `private Rect(){...}`

- A. (1)
- B. (2)
- C. (3)
- D. (3) and (4)

3. If object1 and object2 are objects of the same class, to make object2 a copy of object1, the best approach is:

- A. Use the default constructor to create object2 with object1 data members
- B. Assign object1 to object2 like this: `object2 = object1;`
- C. Write a copy method that will make a field by field copy of object1 data members into object2 data members
- D. Use the copy method that is inherited from the class `Object`

4. Which of the following is indicative of an aggregation relationship between two classes:

- A. Is-A relationship
- B. One class is an inner class of the other
- C. Has-a relationship
- D. One class is a subclass of the other

5. Assume the class `BankAccount` has been created, and the following statement correctly calls a constructor to create an instance of the class:

```
BankAccount account = new BankAccount(5000.0);
```

When the following statement is executed:

```
System.out.println(account);
```

- A. A runtime error will occur
- B. It will print 5000.0
- C. The `toString` method will be called
- D. It will print binary numbers corresponding to the binary representation of the object `account`

6. A method's signature consists of

- A. the method name and the parameter list and their types.
- B. the return type, the method name, and the parameter list and their types.
- C. the method name.
- D. the return type and the method name.

7. Which of the following is true about the static methods

- A. They must be declared public
- B. They can only refer to the static members of the class.
- C. They can only refer to non-static members of the class
- D. An instance of the class should be created to call them.

8. Which of the following is true if the variable *this* is used in a constructor to call another constructor from the same class:

- A. We cannot use *this* in a constructor
- B. It must refers to a parameter in a method
- C. A compiler error will occur if it is before any other line in the method
- D. If it is used in a constructor, it must be the first line

Short Answers (20 points, 5 each)

1. Explain overloading and overriding, use an example to show each one. What is the difference between them?
2. Explain what is meant by polymorphism, provide an example.
3. What are the two characteristics that define an object? What are the accessor methods? Provide an example.
4. What is an interface, how is it different from an abstract class. Provide an example to show.

Problems

1. (30 points, 15 each)

Each person has a name. Employees and students are both persons. Every employee has a title and a salary. Each student has a number for the completed credit hours (an integer). . Employees and students both inherit the abstract method `double makeDonation()` from the class `Person` and implement it specific to themselves. Also `Employee` instances have a method for receiving pay raises which accepts as parameter a percentage and raises the salary by the specified percentage. `Employee` instances implement the `Comparable` interface, this interface has only one method:

```
int compareTo(Employee e) // compares the salary of an employee with another employee's salary and returns 1, 0, or -1.
```

(A) Show a UML class diagram for modeling the above (Hint: make sure to list and indicate everything that is abstract from the not-abstract */

(B) Translate the UML diagram for (A) into Java. Assume that `makeDonation()` returns a fixed amount of 100 for each student, and $0.05 \times \text{salary}$ for each employee.

2. (10 points)

Look at the following classes:

```
public class SubClass1 extends SuperClass1
{
    public SubClass1()
    {
        System.out.println("This is the subclass " + "constructor.");
    }
}

public class SuperClass1
{
    public SuperClass1()
    {
        System.out.println("This is the superclass " + "constructor.");
    }
}
```

What does the following program display (assume the program compiles):

```
public class ConstructorDemo
{
    public static void main(String[] args)
    {
        SubClass1 obj = new SubClass1();
    }
}
```