

*In The Name of God*

# AP Homework 4

Parham Alvani

March 13, 2014

powered by L<sup>A</sup>T<sub>E</sub>X

## Contents

1	Problem 1	2
2	Problem 2	2
3	Problem 3	2
4	Problem 4	2
5	Problem 5	3
6	Problem 6	3
7	Problem 7	3

## 1 Problem 1

A superclasss *public* members are accessible anywhere that the program has a reference to an object of that superclass or to an object of one of its subclasses. When an object of a subclass is instantiated, a superclass *constructor* is called implicitly or explicitly.

Subclass constructors can call superclass constructors via the *super* keyword.

All classes in Java inherit directly or indirectly from the *Object* class.

enum constants are implicitly *public static*.

## 2 Problem 2

Superclass constructors are not inherited by subclasses. **True**

When a subclass redefines a superclass method by using the same signature, the subclass is said to *override* that superclass method. **False**

Inheritance reduces program-development time *and do not* make find bugs more sophisticated. **False**

## 3 Problem 3

BLUEJ PROJECT HAS BEEN INCLUDED

## 4 Problem 4

**1** Polymorphism : In programming languages and type theory, polymorphism is the provision of a single interface to entities of different types. A polymorphic type is a type whose operations can also be applied to values of some other type, or types. There are several fundamentally different kinds of polymorphism:

- If a function denotes different and potentially heterogeneous implementations depending on a limited range of individually specified types and combinations, it is called ad hoc polymorphism. Ad hoc polymorphism is supported in many languages using function overloading.
- If the code is written without mention of any specific type and thus can be used transparently with any number of new types, it is called parametric polymorphism. In the object-oriented programming community, this is often known as generics or generic programming. In the functional programming community, this is often simply called polymorphism.
- Subtyping (or inclusion polymorphism) is a concept wherein a name may denote instances of many different classes as long as they are related by some common superclass. In object-oriented programming, this is often referred to simply as polymorphism.

**2** Constructor overloading : Function overloading or method overloading is a feature found in various programming languages such as Ada, C++, D, and

Java, that allows creating several methods with the same name which differ from each other in the type of the input and the output of the function. It is simply defined as the ability of one function to perform different tasks.

**3 Overriding :** Method overriding, in object oriented programming, is a language feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its superclasses or parent classes. The implementation in the subclass overrides (replaces) the implementation in the superclass by providing a method that has same name, same parameters or signature, and same return type as the method in the parent class. The version of a method that is executed will be determined by the object that is used to invoke it. If an object of a parent class is used to invoke the method, then the version in the parent class will be executed, but if an object of the subclass is used to invoke the method, then the version in the child class will be executed. Some languages allow a programmer to prevent a method from being overridden.

**4 Substitution :** Unification, in computer science and logic, is an algorithmic process of solving equations between symbolic expressions. Depending on which expressions (also called terms) are allowed to occur in an equation set (also called unification problem), and which expressions are considered equal, several frameworks of unification are distinguished. If higher-order variables, that is, variables representing functions, are allowed in an expression, the process is called higher-order unification, otherwise first-order unification. If a solution is required to make both sides of each equation literally equal, the process is called syntactical unification, otherwise semantical, or equational unification, or E-unification, or unification modulo theory.

**5 Casting :** In computer science, type conversion, typecasting, and coercion are different ways of, implicitly or explicitly, changing an entity of one data type into another. This is done to take advantage of certain features of type hierarchies or type representations.

**6 Enum :** In computer programming, an enumerated type (also called enumeration or enum, or factor in the R programming language, and a categorical variable in statistics) is a data type consisting of a set of named values called elements, members or enumerators of the type. The enumerator names are usually identifiers that behave as constants in the language. A variable that has been declared as having an enumerated type can be assigned any of the enumerators as a value. In other words, an enumerated type has values that are different from each other, and that can be compared and assigned, but which are not specified by the programmer as having any particular concrete representation in the computer's memory; compilers and interpreters can represent them arbitrarily.

**7** Abstract Class : In programming languages, an abstract type is a type in a nominative type system which cannot be instantiated directly. Abstract types are also known as existential types. An abstract type may provide no implementation, or an incomplete implementation. Often, abstract types will have one or more implementations provided separately, for example, in the form of concrete subclasses which can be instantiated. It may include abstract methods or abstract properties that are shared by its subtypes.

## **5 Problem 5**

NETBEANS PROJECT HAS BEEN INCLUDED

## **6 Problem 6**

NETBEANS PROJECT HAS BEEN INCLUDED

## **7 Problem 7**

NETBEANS PROJECT HAS BEEN INCLUDED