

Inheritance – Spot questions

Date: 29.02.2024

Marks: 25

1. Define an abstract class **Person with protected fields**: firstName(String), lastName(String), age(int), emailID(String), and constructor with arguments, get, set methods, toString() and
an abstract method : public boolean validateEmail().
Include a private data field: currAddr(Address) that holds the current address
Include method : private void modifyAddress() – that modifies the current address
2. Define a class called **SalesPerson inherited from Person** with members: salarypm(double), totSales(int), empid(int), companyName(String), static field: festivalBonus and methods are private constructor with arguments and public default constructor, set, get methods , and toString()
Include methods : double getBonus() – that returns the bonus obtained based on totSales
double calSal() – that find the annual income of the Salesperson including
void setfest(double) - to initialize the
double getfest() -
Override the validateEmail() which will return true if the email id is firstName followedby empid@companyName.com
3. Modify class **Student to inherit from class Person**, include the following:
University(String) – field that stores the name of the University
Override the validateEmail() which will return true if the email id is first five letters of firstName followed by four letters of lastname@university.edu
4. Define **class DailyWages inherited from class Person** with fields wagesperhr(double), noHrs(int) and constructor with arguments, get, set methods, toString(), double CalaculateWages() that returns the wages as the product of wagesperhr and noHrs .
Make emailID as NULL for DailyWages objects.
5. Define a **Test class** to create objects of SalesPerson, Student and DailyWages and invoke the methods.
 1. Create an array of Person that could hold five objects : Person p[] = new Person[5]
 2. The array can store objects of type of SalesPerson or Student or DailyWages.
 3. invoke the following : p.validateEmail() only if the p holds reference to an object of type SalesPerson or Student
(use instanceof operator to validate the type of object – object instanceof className returns true or false)

NOTE: use existing code for classes Person, SalesPerson, Student and make the necessary changes.