Beni-Suef University

Faculty of Computers and Artificial Intelligence

First Term 2021/2022

CS 241 - Object Oriented Programming

SHEET #5

Question 1:

Write the Customer and Invoice classes which are designed in the following class diagram:

Invoice
-id:int -customer:Customer -amount:double
+Invoice(id:int,customer:Customer,amount:double) +getId():int +getCustomer():Customer +getAmount():double +setCustomer(customer:Customer):void +setAmount(amount:double):void +getCustomerId():int +getAmountAfterDiscount():double +toString():String

Customer
-id:int -name:String -discount:int
+Customer(id:int,name:String,discount:int) +getId():int +getName():String +getDiscount():int +setDiscount(discount:int):void +toString():String

Note:

- The discount in the Customer class is a percentage.
- The method toString in the Customer class should print the output as follows: "name(id)(discount%)"

- The method toString in the Invoice class should print the output as follows:

  Invoice no. : -----

  Customer : name(id)(discount%)

  Amount: ------

  Amount after discount: -------

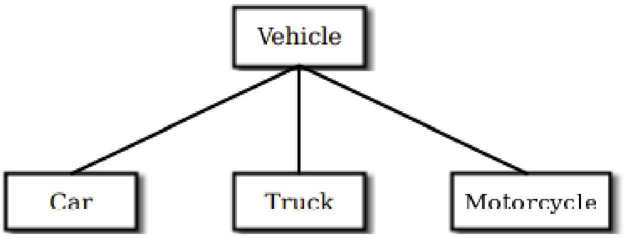
Question 2

Design a class named Person and its two subclasses named Student and Employee. Make Faculty and Staff subclasses of Employee.

- A person has a name, address, phone number, and e-mail address.
- A student has a class status (freshman, sophomore, junior, or senior). Define the status as a constant.
- An employee has an office, salary, and date hired.
- A faculty member has office hours and a rank.
- A staff member has a title.
- Override the toString method in each class to display the class name and the person's name.
- Draw the UML diagram for the classes and implement them.
- Write a test program that creates a Person, Student, Employee, Faculty, and Staff, and invokes their toString() methods.

Question 3:

Suppose that a program has to deal with motor vehicles, the program could use a class named Vehicle to represent all types of vehicles. Since cars, trucks, and motorcycles are types of vehicles, they would be represented by subclasses of the Vehicle class, as shown in this class hierarchy diagram:



Design the class vehicle and its three subclasses considering the following:

- The Vehicle class includes instance variables such as registrationNumber (int) and owner (Person- use class Person in question 2), Speed(int), regularPrice(double), color(String) and two instance methods:
  public void transferOwnership(Person newOwner)
  public double getSalePrice() //returns the regularPrice
- The Car class will add an instance variable numberOfDoors(int).
- The Truck class might have numberOfAxles(int), weight(int), and the method
  public double getSalePrice() //the method checks if the weight>2000, get 10% discount on the regular price. Otherwise, get 20% discount.

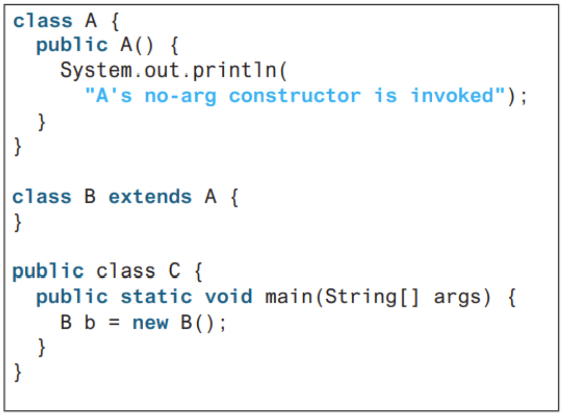
a. The Motorcycle class could have manufacturerDiscount (int), a boolean variable hasSidecar, and the method
public double getSalePrice() //the method subtracts the manufacturerDiscount from the sale price computed from Vehicle class.

- Write a method called printDetails using the following header:
  Public static void printDetails (Vehicle myVehicle)
  to print out relevant data about the Vehicle referred to by myVehicle. The method can receive any type of vehicle as a parameter and prints its details. If it’s a Car, you will want to print out the car’s numberOfDoors and so on.

Question 4:

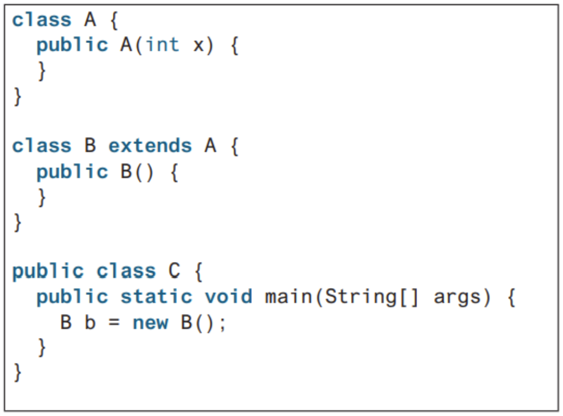
What is the output of running the class C in (a)?

What is the problem in compiling the program in (b)?



```
class A {
  public A() {
    System.out.println(
      "A's no-arg constructor is invoked");
  }
}

class B extends A {
}

public class C {
  public static void main(String[] args) {
    B b = new B();
  }
}
```

(a)

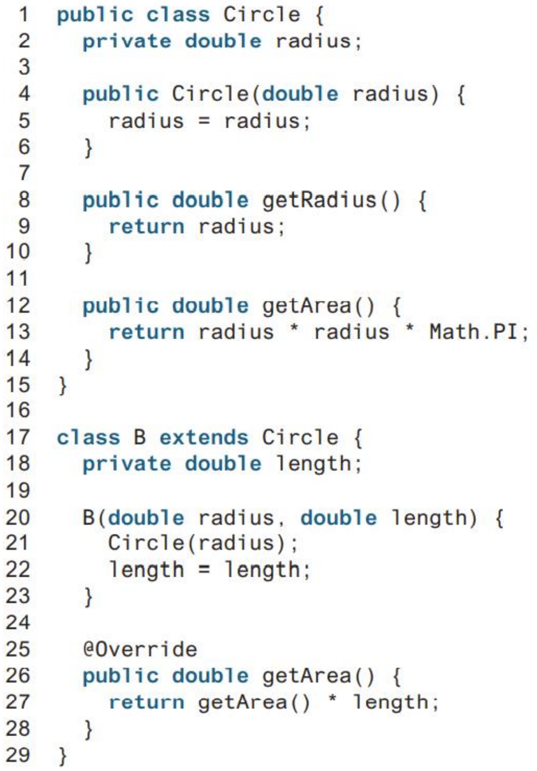


```
class A {
  public A(int x) {
  }
}

class B extends A {
  public B() {
  }
}

public class C {
  public static void main(String[] args) {
    B b = new B();
  }
}
```

(b)

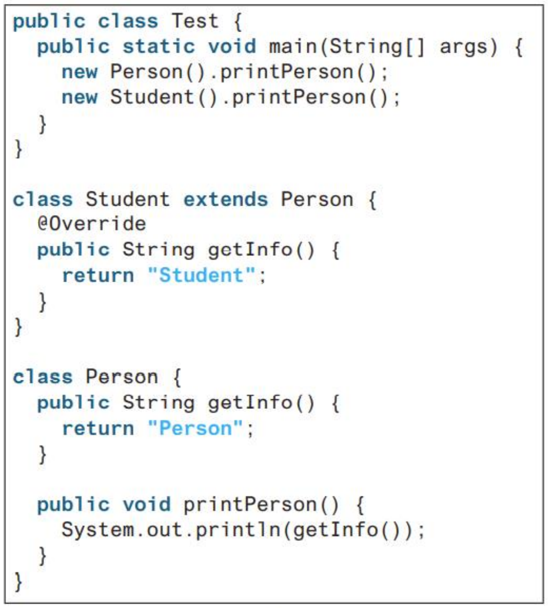
Question 5:

Identify the problems in the following code:



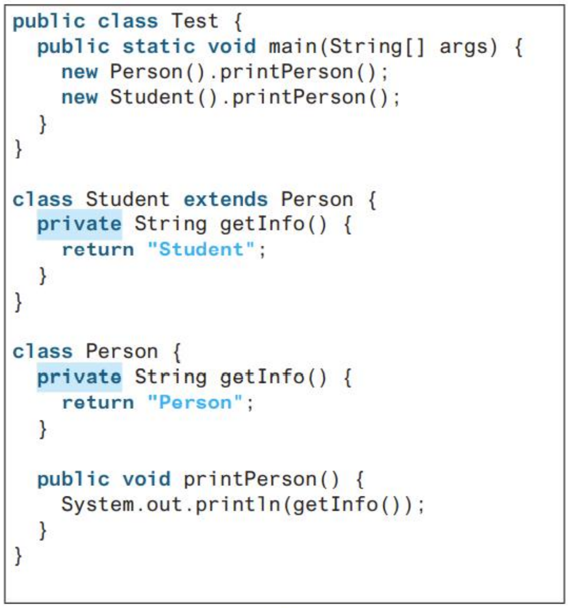
```
public class Circle {
  private double radius;

  public Circle(double radius) {
    radius = radius;
  }

  public double getRadius() {
    return radius;
  }

  public double getArea() {
    return radius * radius * Math.PI;
  }
}

class B extends Circle {
  private double length;

  B(double radius, double length) {
    Circle(radius);
    length = length;
  }

  @Override
  public double getArea() {
    return getArea() * length;
  }
}
```

Question 6: Show the output of the following:



```
public class Test {
  public static void main(String[] args) {
    new Person().printPerson();
    new Student().printPerson();
  }
}

class Student extends Person {
  @Override
  public String getInfo() {
    return "Student";
  }
}

class Person {
  public String getInfo() {
    return "Person";
  }

  public void printPerson() {
    System.out.println(getInfo());
  }
}
```

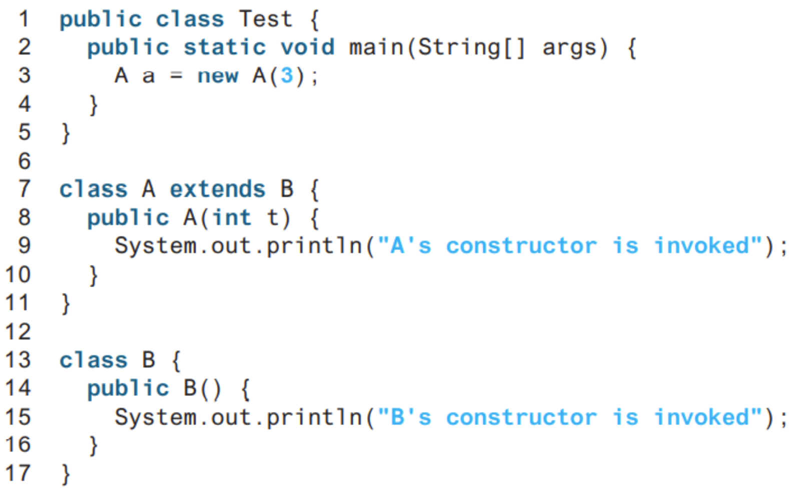
(a)



```
public class Test {
  public static void main(String[] args) {
    new Person().printPerson();
    new Student().printPerson();
  }
}

class Student extends Person {
  private String getInfo() {
    return "Student";
  }
}

class Person {
  private String getInfo() {
    return "Person";
  }

  public void printPerson() {
    System.out.println(getInfo());
  }
}
```

(b)

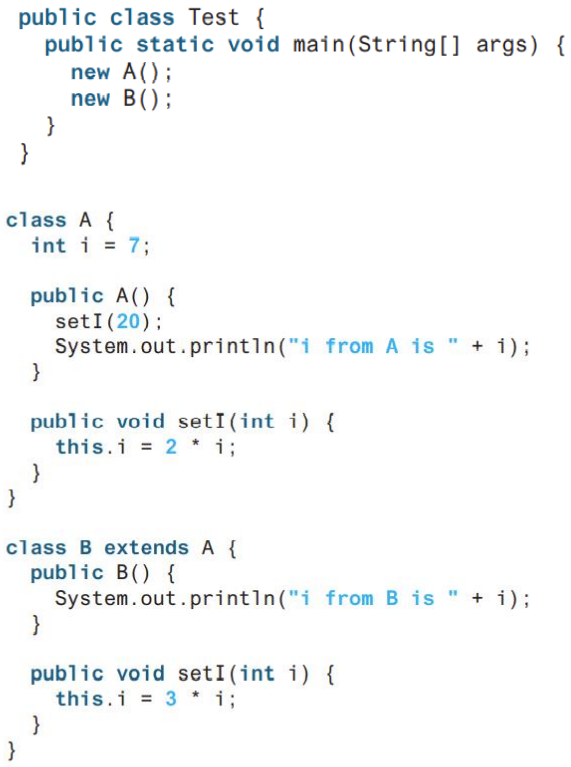
Question 7: Show the output of the following:



```
public class Test {
  public static void main(String[] args) {
    A a = new A(3);
  }
}

class A extends B {
  public A(int t) {
    System.out.println("A's constructor is invoked");
  }
}

class B {
  public B() {
    System.out.println("B's constructor is invoked");
  }
}
```

Is the no-arg constructor of Object invoked when new A(3) is invoked?

Question 8: Show the output of the following:



```
 public class Test {
   public static void main(String[] args) {
     new A();
     new B();
   }
 }


class A {
  int i = 7;

  public A() {
    setI(20);
    System.out.println("i from A is " + i);
  }

  public void setI(int i) {
    this.i = 2 * i;
  }
}

class B extends A {
  public B() {
    System.out.println("i from B is " + i);
  }

  public void setI(int i) {
    this.i = 3 * i;
  }
}
```

Question 9:

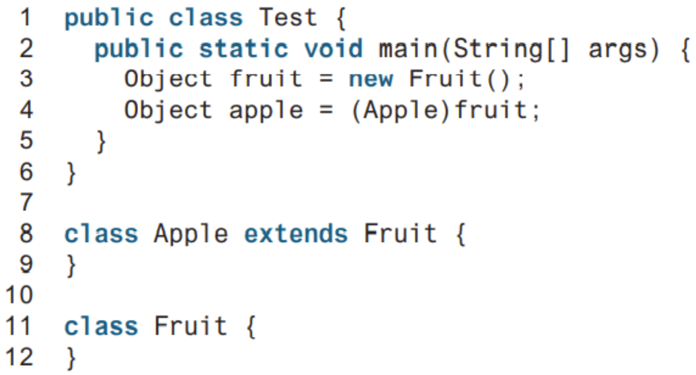
For the Shape and Circle classes, answer the following questions:

a. Assume that circle and object1 are created as follows:
Circle = new Circle();
Shape object1 = new Shape();
Are the following Boolean expressions true or false?
- (circle instanceof Shape)
- (object1 instanceof Shape)
- (circle instanceof Circle)
- (object1 instanceof Circle)

b. Can the following statements be compiled?
Circle circle = new Circle();
Shape object1 = circle;

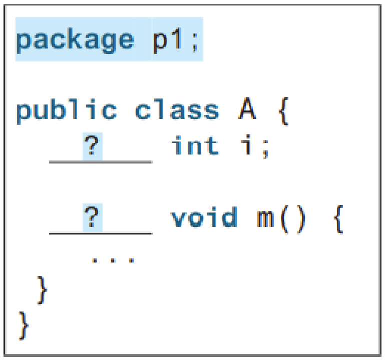
c. Can the following statements be compiled?
Shape object1 = new Shape();
Circle circle = (Circle)object1;

Question 10: What is wrong in the following?



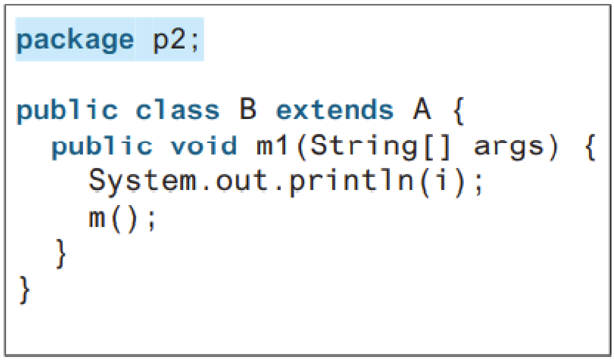
```
public class Test {
  public static void main(String[] args) {
    Object fruit = new Fruit();
    Object apple = (Apple)fruit;
  }
}

class Apple extends Fruit {
}

class Fruit {
}
```

Question 11:



```
package p1;

public class A {
   __?__  int i;

   __?__  void m() {
     ...
   }
}
```

(a)



```
package p2;

public class B extends A {
  public void m1(String[] args) {
    System.out.println(i);
    m();
  }
}
```

(b)

Show if class B can be compiled if we replace the question marks in class A by:

- Private
- Blank
- Protected

Question 12: Indicate true or false for the following statements:

a. A protected datum or method can be accessed by any class in the same package.

b. A protected datum or method can be accessed by any class in different packages.

c. A protected datum or method can be accessed by its subclasses in any package.

d. A final class can have instances.

e. A final class can be extended.

f. A final method can be overridden.

Best Wishes
DR. Noha Yehia