Petra University *(Private Accredited University)* Faculty of Information Technology Department of Computer Science	جامعة البترا	جامعة البترا (جامعة خاصة معتمدة) كلية تكنولوجيا المعلومات قسم علم الحاسوب
Course Title:Programming Language II	جامعة البترا	اسم الطالب:
Instructor Name: Huda Saadeh, Khalil Omar, Abed Al-Kareem Al-Banna	**Final Exam**	الرقم الجامعي:
Date:26/1/2012		الشعبة:
Time:13:30-15:30	2011-2012 Final Exam	الدرجة: المجموع: / 50

Theoretical Part

Q1) Multiple Choices [25 Points]:
Note there are 3 bonus questions.

Practical Part

Q2) Analyze the UML in <u>figure1</u> and answer 10 of the following [5 Points]:

1. An example of a composition: Owners in Cars class.
2. An example of an aggregation: fines in TicsNotice.
3. owner is a subclass.
4. An example of a weak is_a relation: _____.
5. An example of a has_a relation: _____.
6. _____ is a package private (default access modifier) variable.
7. An example of interface : _____
8. _____ is a private variable.
9. _____ is a static variable.
10. _____ is a protected variable / method.
11. _____ is a abstract method.
12. An example of generic programming: _____.
13. _____ is an example of Data Binding.
14. Root (Super) of all classes in the system: _____.

Q3) Figure1 shows the UML for traffic fines system keeps track of fines types, cars and their owners. This system helps drivers to know their fines and total amounts that should be paid. [20 Points]

- Implement the Traffic System UML [2].
- Add overriding *public Boolean equals(Object O)* method into Owner class [1].
- Add overriding *public String toString()* method into Owner class [1].
- Add overriding *protected Object clone() throws CloneNotSupportedException* method into Cars class [1].
- Cars class implements the Cloneable interface [1]:

```
public interface Cloneable {
}
```

- Add an abstract method *void printInfo();* into fines class [2].
- GeneralFines class implements comparable interface to compare points amounts only [2]:

```
public interface Comparable {
  public int compareTo(Object o);
}
```

- Write the code of FinesNotice interface [2]:
  - When implementing **totalAmount** method it should return the summation of finesAmount for each TakingFines object.
  - When implementing **add** method it should insert a Fines object into the fn array and increment the fnSize and noOfFines by one.
- Write the code of TakingFines class knowing that[6]:
  - fn[] is an array of Fines objects.
  - Cr is a Car object
  - fnSize is an integer variable that keeps the number of objects in fn array.
  - fnSize is an immutable data field.
  - noOfFines is an integer variable that keeps the number of fines for all cars (shared by all TakingFines objects).
- Add user_defined constructor for TakingFines and Fines classes [2].