BIRKBECK

(University of London)

MSc EXAMINATION FOR INTERNAL STUDENTS

MSc Computer Science
MSc Data Science

Department of Computer Science and Information Systems

Principles of Programming II

BUCI064H7

DATE OF EXAMINATION: Wednesday, 2nd May 2018
DURATION OF PAPER: One hour

Mock written paper

WITH OUTLINE SOLUTIONS

RUBRIC:

1. Candidates should attempt ALL 6 questions on this paper.
2. You are advised to look through the entire examination paper before getting started, in order to plan your strategy.
3. Simplicity and clarity of expression in your answers is important.
4. All programming questions should be answered using the JAVA programming language.
5. Electronic calculators are **NOT** allowed.
6. START EACH QUESTION ON A NEW PAGE.

Question:	1	2	3	4	5	6	Total
Marks:	6	6	7	9	8	14	50

Question 1 .... Total: 6 marks

Briefly explain the following terms:

(a) abstract class, **3 marks**

(b) interface. **3 marks**

Use appropriate examples to illustrate your answer. Include in your answers the use of the keywords `extends` and `implements`, and indicate where they are (and are not) applicable.

Solution: An abstract class is one that is not used to construct objects, but only as a basis for making subclasses.

An interface consists of a set of instance method interfaces, without any associated implementations. (Can also contain constants.)

Should use `extends` for extending an interface or concrete class; `implements` for instantiation of an interface as a (possibly abstract) class.

Question 2 .... Total: 6 marks

How does method overloading differ from method overriding? Provide appropriate examples to illustrate your answer.

Solution: Roughly three marks each including appropriate examples.

method overloading same method name with a different signature.

method overriding method with same signature replaced in subclass.

Question 3 .... Total: 7 marks

This question concerns Unit testing and TDD.

(a) What is Unit testing? Provide a brief example to illustrate your answer. **3 marks**

Solution: A unit test is the smallest testable part of an application like functions, classes, procedures, interfaces. Unit testing is a method by which individual units of source code are tested to determine if they are fit for use.

(b) State four advantages of carrying out Unit testing. **4 marks**

Solution: Something like the following but other answers maybe equally acceptable.

1. Issues are found at early stage. Since unit testing are carried out by developers where they test their individual code before the integration. Hence the issues can be found very early and can be resolved then and there without impacting the other piece of codes.

2. Unit testing helps in maintaining and changing the code. This is possible by making the codes less interdependent so that unit testing can be executed. Hence chances of impact of changes to any other code gets reduced.

3. Since the bugs are found early in unit testing hence it also helps in reducing the cost of bug fixes. Just imagine the cost of bug found during the later stages of development like during system testing or during acceptance testing.

4. Unit testing helps in simplifying the debugging process. If suppose a test fails then only latest changes made in code needs to be debugged.

Question 4 . . . . Total: 9 marks

For each of the following statements indicate whether it is *true* or *false*.
There is one mark for each correct answer.
Please note: Incorrect answers will be penalised.

(a) **F** A public top-level Java class may be defined in a source file with any base name as long as the file extension is `.java`.

(b) **F** Java identifiers can contain letters, digits, and the underscore symbol and may start with a digit.

(c) **F** The statement `int x = 3f/4f;` will compile, but the result will be truncated so that `x` gets the value `0`.

(d) **F** In a for loop header, `for (`*initialiser*`;` *condition*`;` *update*`)`, the Java compiler requires *initialiser* to initialise a loop variable and *update* to update it.

(e) **T** The declarations `double[] scores` and `double... scores` are equivalent.

(f) **F** Java arrays have a variable number of elements of mixed types.

(g) **F** Given an array named `scores`, the statement `scores[scores.length + 1]` will not compile.

(h) **F** Instance methods of a class can be called without first instantiating an object of the class.

(i) **F** Every Java class has a default no-arg constructor in addition to the constructors you write yourself.

Solution: ½ mark deducted for an incorrect answer with a minimum of zero overall.

Question 5 . . . . Total: 8 marks

Consider the following two classes:

```
public class Base {
        public void methodOne() {
                System.out.println("A");
                methodTwo();
        }

        public void methodOne(int a) {
                System.out.println("W");
                methodTwo();
        }

        public void methodTwo() {
                System.out.print("B");
        }
}
```

```
public class Derived extends Base {
        public void methodOne(int a) {
                super.methodOne();
                System.out.print("X");
        }

        public void methodOne(Integer a) {
                super.methodOne();
                System.out.print("C");
        }

        public void methodTwo() {
                super.methodTwo();
                System.out.print("D");
        }
}
```

The following declaration appears in a client program.

```
    Base b = new Derived();
```

What is printed as a result of the call `b.methodOne(0)`? You should include a trace of the execution of the code.

Solution: 2 marks for getting A
2 marks for getting B
1 mark for X
3 marks for D

```
A
BDX
```

Question 6 . . . . Total: 14 marks

You are given the following class definitions:

```java
package question;

public abstract class GrandParent {
        private final static int HowMuch;

        static {
                System.err.println("jkahdjkashd");
                HowMuch = 100;
        }

        public abstract boolean good();

        public static int getMore() {
                return HowMuch;
        }

        public void eat() {
                if (good())
                        System.out.println("Good!");
                else
                        System.out.println("Bad!");
        }
}
```

```java
package question;

public class Parent extends GrandParent {
        @Override
        public boolean good() {
                System.out.println(getAge());
                System.out.println(getMore());
                return getAge() < getMore();
        }

        public int getAge() {
                return 42;
        }

        @Override
        public void eat() {
                super.eat();
                System.out.println("Ahh.........!");
        }
}
```

```java
package question;

public class Child extends Parent {

        @Override
        public int getAge() {
                return super.getAge() / 2;
        }

        public static int getMore() {
                return 30;
        }
}
```

```java
package question;

public class Test {
        public static void main(String[] args){
                Child c = new Child();
                c.eat();
        }
}
```

(a) What is the significance of `static` in the declaration of the `getMore` method in the class `GrandParent`? 2 marks

Solution: Class level method. No object required to invoke the method. Compile time binding.

(b) What is the significance of the keyword `abstract` in the declaration of the method `good` in the class `GrandParent`? 2 marks

Solution: Required to be implemented by subclasses.

(c) What output is produced if we execute the `main` method of the `Test` class? You should explain your answer clearly. 10 marks

Solution:

```
21
100
Good!
Ahh.........!
```