School of Business and Information Technology

IT7374 Programming III

Assignment 1 – Tri 1 2016

Due Date:	**22 April 2016 (11:55pm)**
Total marks:	**100**
Weighting:	**30% of final mark**

Assessment Guidelines

- You are required to submit an electronic copy of your source code and compiled byte codes.
- Create a project folder called **IT7374_Assignment1_yourStudentID** where you will place all your files. Remember to zip this folder before submitting it to Assignment 1 submission box in the IT7374 Moodle course page.
- Submit your work on or before the due date. The submission box will be closed and will no longer be accessible after the deadline.

Late Work & Extension

- Handing in the work after the due date without extension is considered **late**. Students will be advised of any penalties to be imposed for late submission.
- Lecturers reserve the right not to mark work that is handed in late.
- Students are encouraged to manage their schedules in order to meet deadlines. Under exceptional cicumstances (e.g. bereavement, illness, accident) students may be granted an extension of the due date of Assignment 1.
- Extensions are **not** granted automatically. A request for extension must be made at least 24 hours before the deadline of Assignment 1. If a student applies for an extension, a relevant evidence (e.g. medical certificate) must be presented.

Pass Requirements

- Students must attempt all Assignments and Exam.
- Students must obtain at least 40% in the examination and a final mark average of 50%.

Summary of Tasks for Assignment 1

For this assignemnt, you are required to show your competency in applying some of the advanced features of Object-Oriented Programming in Java such as:

- manipulating classes at runtime
- designing and implementing multi-threaded application program
- implementing and discussing object persistence using appropriate mechanism

Task 1 [30 marks]

As we know, a Java `.class` file is a file that is produced by the compiler from Java source file (`.java` file). It contains the Java bytecode of a program that it can be executed on the JVM. Your task is to use the **Java reflection API** to generate `.java` source files from the `.class` file of any original class of your choice.
The class that is used must:

1. Be derived from any original abstract or concrete base class.
2. Implement an original interface.

Your application should generate an exact copies of the original Java files excluding:

1. Attribute and argument names, and
2. Logic within methods!

Create a separate package for Task1 and include it in the IT7374_Assignment1_yourStudentID folder.

Task 2 [40 marks]

Design and implement a multiple threaded application in Java. Your program should contain two or more parts that are running concurrently and each part can handle different tasks at the same time. Each thread should be able to run in parallel with the other threads.

Include your source files to your project folder IT7374_Assignment1_yourStudentID.

Write a program documentation making sure that the following items are included:

1. Description of the scenario that you chose to implement the multithreading functionality.
2. List of all classes, methods, objects, interfaces, etc. that are used in your program.
3. Explanation as to how your program achieved a multithreading functionality. Also identify the activities that can proceed simultaneously during execution.
4. Discussion as to why you implemented Synchronization to your objects.
5. Provide a screenshot illustrating the execution of your program.

Task 3 [30 marks]

Write a report (approx. 3 pages in length) describing the Java *clone* mechanism. Your report should cover:

a) A discussion on why you think the clone mechanism is required by the Java language.
b) A discussion and illustration by providing an example on how the clone mechanism is implemented in Java.
c) An explanation about the different options available to developers for implementing cloning when cloning inheritance hierarchies.
d) Some advantages and disadvantages of the clone mechanism.
e) Suggested alternatives to the Java clone mechanism.

Assignment Requirements

Below is your checklist to be completed before you submit your Assignment 1. Assignment 1 is consists three (3) tasks that must all be completed to obtain full marks.

Task 1 – Java Reflection API	Completed
▪ A selection of class that is derived from an original abstract or concrete base class.	
▪ The selected class must implement an original interface.	
▪ The application should use Java Reflection API to generate `.java` source files from the `.class` file of the original class.	
▪ The application should be able to generate exact copies of the original Java files excluding ➢ Attribute and argument names ➢ Logic within the methods	
Task 2 – Multi-threading Application in Java	
▪ The working multi-threaded application: ➢ Contains two or more parts that can run concurrently	
➢ Each part can handle different tasks at the same time	
➢ Each thread runs in parallel with the other threads	
▪ The documentation includes: ➢ Description of the chosen scenario	
➢ Name of the classes, methods, and objects that exists in the program	
➢ Explanation as to how multithreading functionality is achieved	
➢ Discussion on why you implemented Synchronization to your objects	
➢ Screenshot illustrating the execution of the program	
Task 3 – Cloning Mechanism in Java	
▪ Gave the reason/s as to why cloning mechanism is required in Java	
▪ Explained how clone mechanism is implemented	
▪ Identified the different options available to developers for implementing cloning when cloning inheritance hierarchies	
▪ Identified some advantages and disadvantages of clone mechanism	
▪ Named some alternatives to Java cloning mechanism	

IT7374 – Programming III

Assignment 1 Marking Scheme

Task 1 – Java Reflection API	**Marks**
▪ A selection of class that is derived from an original abstract or concrete base class.	
▪ The selected class must implement an original interface.	
▪ The application should use Java Reflection API to generate `.java` source files from the `.class` file of the original class.	
▪ The application should be able to generate exact copies of the original Java files excluding ➢ Attribute and argument names ➢ Logic within the methods	
Total	**30**
Task 2 – Multi-threading Application in Java	
▪ The working multi-threaded application: ➢ Contains two or more parts that can run concurrently	
➢ Each part can handle different tasks at the same time	
➢ Each thread runs in parallel with the other threads	
▪ The documentation includes: ➢ Description of the chosen scenario	
➢ Name of the classes, methods, and objects that exists in the program	
➢ Explanation as to how multithreading functionality is achieved	
➢ Discussion on why you implemented Synchronization to your objects	
➢ Screenshot illustrating the execution of the program	
Total	**40**
Task 3 – Cloning Mechanism in Java	
▪ Gave the reason/s as to why cloning mechanism is required in Java	
▪ Explained how clone mechanism is implemented	
▪ Identified the different options available to developers for implementing cloning when cloning inheritance hierarchies	
▪ Identified some advantages and disadvantages of clone mechanism	
▪ Named some alternatives to Java cloning mechanism	
Total	**30**
Overall Total	**100**