

5 Further Java (arb33)

Clients of an online service execute code remotely at runtime by sending and receiving serialised Java objects to and from a server. Clients use the following classes:

```
public class NewClass implements Serializable {
    private String name;
    private byte[] bytecode;
}

public class Result implements Serializable {}
public class Params implements Serializable {}

abstract public class Function<R extends Result, P extends Params>
    implements Serializable {
    abstract public R run(P args);
}

class Invoke implements Serializable {
    private Function function;
    private Params args;
}
```

Clients send the bytecode of new classes using `NewClass` and an instance of `Invoke` to request the server execute a specified function. The server responds to such a request by executing the `run` method in the `Function` object and serialising and returning the result back to the client. You may assume the above classes have appropriate constructors as well as getter and setter methods.

- (a) Define subclasses of `Params`, `Result` and `Function` (called `Squared`) to compute x^2 where x is of type `double`. [3 marks]
- (b) Implement a client which executes `Squared`. Your solution should accept an argument x from the user, remotely execute the code and print out the result. You may assume a static method `byte[] getClassBytes(Class c)`. [5 marks]
- (c) An `Invoke` object can contain incompatible `Function` and `Params` references. Provide an example and describe how to refactor your client to include a compile-time compatibility check. Can the server also check? [5 marks]
- (d) Describe the security challenges in the server implementation. How might these be addressed? [3 marks]
- (e) Describe how you could improve runtime error handling by the server as well as reporting such errors to the client. [4 marks]