Q5

* Required

1. **Surname and Name and ID** *

.....

2. **What was the problem with collections that motivated the introduction of generics?**
Mark only one oval.

- ( ) The types of the objects in the collections were intermixed
- ( ) The types of the objects in the heap was not clear
- (x) Getting from collections required casting
- ( ) Getting from collections required compiler checks

3. **What is typer erasure in the following example?**
There are ArrayList<Integer> and ArrayList<Natural>
Mark only one oval.

- (x) There is only one .class file with name ArrayList.class
- ( ) There are two .class files with name ArrayList.class
- ( ) There are two .class files one is included in the other as Natural is parent class of Integer
- ( ) There are two .class files with name Natural.class

4. **What is collection of unknown?**
Mark only one oval.

- ( ) The child class of any type
- ( ) The child class of any collection
- ( ) The parent class of any generic collection with type Object
- (x) The parent class of any generic collection

5. **Which of these is a bounded wildcard?**
Mark only one oval.

- ( ) ArrayList<Shape>
- (x) ArrayList<? extends Shape>
- ( ) ArrayList<?>
- ( ) ArrayList<T> T a parameter

6. **What is type inference?**
When types in parametrised methods are different,
Mark only one oval.

(x) the compiler infers the most generic type
( ) the compiler infers the most specialised type
( ) the compiler infers the most specific type
( ) the compiler throws an error

7. **What is the Liskov principle?**
A is parent class of B
Mark only one oval.

(x) Any object of type A can be substituted with an object of type B
( ) Any object of type B can be substituted with an object of type A
( ) Any object of type A can be substituted with an object of type B only if A is abstract
( ) Any object of type A can be substituted with an object of type B only if B has some invariants

8. **What is true?**
Mark only one oval.

( ) Specialisation extends a child class by overriding parent class's methods
( ) Specification extends a child class with new functionalities
( ) Overriding is performed to extend the functionality of a child class
(x) Extension for limitation throws exceptions for those methods that we do not want in the child class

9. **Which of the following is true?**
A is parent class of B. A myA = newB(); f is a static method both in A and B
Mark only one oval.

( ) myA.f() is compiler error
( ) myA.f() is run-time error
(x) myA.f() calls the method in A
( ) myA.f() calls the method in B

10. **What is delegation?**
A and B are two classes. B has a method move().
Mark only one oval.

( ) Overriding the method move() in A
( ) Overriding the method move() in B
(x) Creating a method move() in A which instantiates an object of type B and calls the method move through this object
( ) B satisfies the Liskov principle

11. **What parametrised generics are useful for?**
Mark only one oval.

- (x) We can finally wrap anonymous objects in generic collections
- ( ) We can finally compile generic collections
- ( ) We can finally run generic collections
- ( ) We can finally get an object of unknown type from collections