Started on	Tuesday, 24 September 2024, 8:32 PM
State	Finished
Completed on	Tuesday, 24 September 2024, 8:52 PM
Time taken	20 mins 1 sec
Marks	0.00/10.00
Grade	**0.00** out of 100.00

Question **1**

Not answered

Marked out of 1.00

Given the following classes, which of the following can independently replace INSERT IMPORTS HERE to make the code compile? (Choose all that apply)

```
package aquarium;
public class Tank { }
package aquarium.jellies;
public class Jelly { }
package visitor;
INSERT IMPORTS HERE
public class AquariumVisitor {
public void admire(Jelly jelly) { } }
```

Select one or more:

☑ **a. import aquarium.jellies.*;**

☑ **b. import aquarium.jellies.Jelly;**

□ **c. import aquarium.*;**

□ **d. import aquarium.jellies.Jelly.*;**

□ **e. None of these can make the code compile.**

□ **f. import aquarium.*.Jelly;**

Question **2**

Not answered

Marked out of 1.00

What is the output of the following application?

```
1: public class CompareValues {
2: public static void main(String[] args) {
3: int x = 0;
4: while(x++ < 10) {}
5: String message = x > 10 ? "Greater than" : false;
6: System.out.println(message+","+x);
7: }
8: }
```

Select one or more:

- □ **a. false,11**
- □ **b. The code will not compile because of line 4.**
- □ **c. Greater than,10**
- ☑ **d. The code will not compile because of line 5.**
- □ **e. false,10**
- □ **f. Greater than,11**

Question **3**

Not answered

Marked out of 1.00

Which are true statements? (Choose all that apply)

Select one or more:

- □ **a. An immutable object cannot be garbage collected.**
- ☑ **b. An immutable object cannot be modified.**
- ☑ **c. String is immutable.**
- □ **d. An immutable object can be modified.**
- ☑ **e. An immutable object can be garbage collected.**
- □ **f. StringBuffer is immutable**
- □ **g. StringBuilder is immutable**

Question **4**

Not answered

Marked out of 1.00

Which of the following methods compile? (Choose all that apply)

Select one or more:

- □ **a. public int methodG() { return null;}**
- ☑ **b. public void methodA() { return;}**
- ☑ **c. public int methodD() { return 9;}**
- □ **d. public int methodE() { return 9.0;}**
- □ **e. public int methodF() { return;}**
- □ **f. public void methodB() { return null;}**
- ☑ **g. public void methodD() {}**

Question **5**

Not answered

Marked out of 1.00

Which statement(s) are correct about the following code? (Choose all that apply)

```
public class Rodent {
protected static Integer chew() throws Exception {
System.out.println("Rodent is chewing");
return 1;
}
}
```

static is missing here

parent is returning int but here it is returning number

```
public class Beaver extends Rodent {
public Number chew() throws RuntimeException {
System.out.println("Beaver is chewing on wood");
return 2;
}
}
```

Select one or more:

- □ **a. It fails to compile because the method is protected in the parent class and public in the subclass.**
- □ **b. It fails to compile because the type of the exception the method throws is a subclass of the type of exception the parent method throws.**
- □ **c. It will compile without issue.**
- ☑ **d. It fails to compile because of a static modifier mismatch between the two methods.**
- ☑ **e. It fails to compile because the return types are not covariant.**

Question 6

Not answered

Marked out of 1.00

Consider the following Java code that uses a custom object as a key in a HashMap:

```
import java.util.HashMap;
class Person {
    String name;
    int age;

    Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
}

public class Main {
    public static void main(String[] args) {
        HashMap<Person, String> map = new HashMap<>();
        Person p1 = new Person("Alice", 30);
        Person p2 = new Person("Alice", 30);
        map.put(p1, "Engineer");
        System.out.println(map.get(p2));
    }
}
```

What will be the output of the above code?

Select one or more:

□ **a. Compilation error**

□ **b. "Engineer"**

□ **c. Runtime exception**

☑ **d. null**

Question **7**

Not answered

Marked out of 1.00

Consider the following Java code:

```
import java.util.HashMap;


public class Main {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("a", 1);
        map.put("b", 2);
        map.put("a", 3);


        System.out.println(map.get("a"));
        System.out.println(map.size());
    }
}
```

What will be the output?

Select one or more:

□ **a. 1 and 3**

☑ **b. 3 and 3**

□ **c. 1 and 2**

□ **d. 3 and 2**

Question **8**

Not answered

Marked out of 1.00

Consider the following code:

```
import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;


public class Main {

    public static void main(String[] args) {

        HashMap<String, Integer> map = new HashMap<>();

        map.put("one", 1);

        map.put("two", 2);

        map.put("three", 3);


        for (Map.Entry<String, Integer> entry : map.entrySet()) {

            if (entry.getKey().equals("two")) {

                map.remove("two");

            }

        }


        System.out.println(map);

    }

}
```

What will happen when this code is executed?

Select one or more:

□ **a. It will remove the key "two" and print {"one"=1, "three"=3}.**

□ **b. It will run successfully and print {"one"=1, "three"=3} without any errors.**

☑ **c. It will throw a ConcurrentModificationException.**

□ **d. It will remove the key "two" but print the original map.**

Question **9**

Not answered

Marked out of 1.00

What will be the output of the following code?

```
import java.util.HashMap;

public class Main {
    public static void main(String[] args) {
        HashMap<String, String> map = new HashMap<>();
        map.put(null, "value1");
        map.put("key1", null);
        map.put(null, "value2");

        System.out.println(map.get(null));
        System.out.println(map.get("key1"));
    }
}
```

Select one or more:

□ **a. "value1" and "value1"**

□ **b. null and null**

☑ **c. "value2" and null**

□ **d. null and "value1"**

Question **10**

Not answered

Marked out of 1.00

Which of the following statements is true about handling collisions in a HashMap?

Select one or more:

□ **a. HashMap uses a linear probing mechanism for collision resolution.**

☑ **b. HashMap uses chaining, where each bucket contains a linked list of entries.**

□ **c. HashMap does not handle collisions.**

□ **d. HashMap doubles its size every time a collision occurs.**