```
Question 1
```

Correct Mark 1.00 out of 1.00

Flag
 question

```
public class ArithmeticOperation(
    private void add(int operand1, int operand2) {
        System.out.println(operand1 + operand2);
        }
    }
    public class Addition extends ArithmeticOperation {
        public void show() {
            add(10, 12);
        }
        public static void main(String args[]) {
            Addition ob = new Addition();
            ob.show();
        }
    }

What will be the output of above code when compiled and executed?

Select one:
    Runtime error as add method is not defined in MethodOverriding class
        Will compile and display 1012
        Will print false
        Will compile and display 32
        Compile time error ✓
```

Question **2**

Correct Mark 1.00 out of 1.00

▼ Flag

question

```
What will be the output of the following program?
class FourWheeler
 public FourWheeler()
   System.out.println("Class FourWheeler");
class Car extends FourWheeler
 public Car()
   System.out.println("Class Car");
class Audi extends Car
 public Audi()
   super();
   System.out.println("Class Audi");
class Driver
 public static void main(String args[])
   Audi cc=new Audi();
Select one:
    Class Audi
    Class Car
    Class FourWheeler
    Compile Time Error
    Exception occurs
    Class FourWheeler
    Class Car
    Class Audi 🗸
```

```
Question
3
```

Correct Mark 1.00 out of 1.00 ▼ Flag

question

```
10. class FourWheeler {
11. private void display() {}
13. class Car extends FourWheeler {
14. protected ✓ void display() {}
15.}
Which method at line 14, will correctly complete class Car?
```

Question 4

Correct Mark 1.00 out of 1.00

▼ Flag

```
Given:
class FourWheeler
public FourWheeler ()
System.out.print(1);
class Car extends FourWheeler
public Car()
System.out.print(2);
class Audi extends Car
public Audi()
System.out.print(3);
public class Driver
public static void main( String[] argv )
new Audi();
What is the result when this code is executed?
Select one:
 O The code runs with no output
 321
 O 3
 123
```

Question 5

Correct Mark 1.00 out of 1.00

Flag question

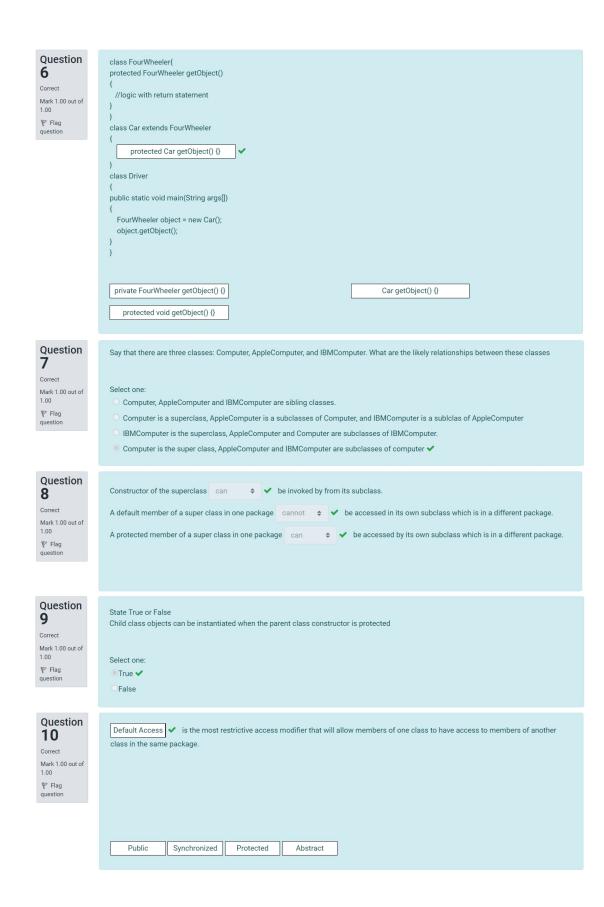
The class Employee is declared in a package mainpack and the Derived classes PermanentEmployee and TemporaryEmployee are declared $in another package \, \textbf{subordpack}. \, The \, basic Pay \, attribute \, should \, be \, accessed \, only \, by \, means \, of \, a \, derived \, class \, object.$

How to ensure that the basicPay attribute is not accessed directly by the other classes in the subordpack?

```
package mainpack ♦ ✔;
public class Employee{
protected ♦ ✓ int basicPay;
/*PermanentEmployee.java*/
```

Employee.java

```
package subordpack ♦ ✔;
public class PermanentEmployee extends Employee{
/\!/ Temporary Employee.java
public class TemporaryEmployee extends Employee{
}
```



Question 11 Correct Mark 1.00 out of 1.00 Flag question

Question 12

Correct Mark 1.00 out of 1.00

▼ Flag
question

Interpret which of the following statements are correct with respect to inheritance relationship in java?

Select one or more:

- $\hfill \Box$ object of subclass referenced by super class type can access newly defined sub class variables
- $^{ extstyle e$
- ☑ object of subclass referenced by super class type can invoke super class methods ✔
- $\hfill \Box$ object of subclass referenced by super class type can invoke newly defined sub class methods
- $\ ^{f ext{$arphi}}$ object of subclass referenced by super class type can access super class variables $\ ^{f ext{$arphi}}$