Question 1

Correct

Mark 1.00 out of 1.00

Flag question

Predict the output

```
abstract class Vehicle
{
   abstract void calcPremium(Number N);
}
interface Insurance
{
   abstract void calcPremium (Object O);
}
class Car extends Vehicle implements Insurance
{
            public void calcPremium (Object O)
              {
                 System.out.println("Object");
              }
   void calcPremium (Number N)
   {
      System.out.println("Number");
   }
}
public class Test
{
   public static void main(String[] args)
   {
      Vehicle a = new Car();
       a. calcPremium (new Integer(121));
        Insurance b = new Car();
         b. calcPremium (new Integer(121));
         Car c = new Car();
          c. calcPremium (new Integer(121));
   }
}
```

Select one:

- Run time error
- Compile time error
- Number
  Object
  Number ✔
- Number
  Number
  Object

Question 2

Correct

Mark 1.00 out of 1.00

Flag question

The type Vehicle has drive functionality. The classes Car and Bike implements the drive functionality and can be further subclassed. Fill in the given code with appropriate access specifier so that the subclasses of Car and Bike do not modify the Drive functionality.

```
interface Vehicle{
  void drive();
}
class Car implements Vehicle{
    public ✔ final ✔ void drive()  {
     //drive  }
}
```

Question 3

Correct

Mark 1.00 out of 1.00

Flag question

If a class extends two interfaces and they both have a method with same name and signature but different return type, then a conflict will arise because the compiler will not able to link a method call due to ambiguity. State true or false.

Select one:

- True ✔
- False

Question 4

Correct

Mark 1.00 out of 1.00

Flag question

11. public interface Status {

12. public static final ✔ double PI = 3.14;

13. }

Fill the correct choice.

Question 5

Correct

Mark 1.00 out of 1.00

Flag question

An interface can contain public, static, final fields (i.e., constants) default and static methods with bodies True ✔

An instance of interface can be created. False ✔

A class can implement multiple interfaces. True ✔

Many classes can implement the same interface. True ✔

Question 6

Correct

Mark 1.00 out of 1.00

Flag question

Predict the output

```
class Car implements Insurance
{
   public int calcPremium(int i)
   {
      return i = i * i;
   }
}
interface Insurance
{
   int calcPremium(int i);
}
public class MainClass
{
   public static void main(String[] args)
   {
      Insurance b = new Car();
      System.out.println(b.calcPremium(2));
   }
}
```

Select one:

- Run time Error
- Compile time error because you cannot create an object of type interface Insurance
- Compile time error because you must create interface before implementing it.
- The output will be 4 ✔

Question 7

Correct

Mark 1.00 out of 1.00

Flag question

Predict the output:

```
interface Employee
{
    int a=90;
}
class PermanentEmployee implements Employee
{
    public void f1()
    {
        a=10;
    }
}
```

Select one:

- error, since variable a is default
- no error
- error, since interfaces Employee is not public
- error, since variable a is assigned a value ✔

Question 8

Correct

Mark 1.00 out of 1.00

Flag question

```
public abstract interface Insurance{
public void insuranceDescription(String s);
}
```

Which is the correct class?

Select one:

- public class Car implements Insurance
  {
  public void insuranceDescription (Integer i) { }
  }
- public class Car extends Insurance
  {
  public void insuranceDescription (Integer i) { }
  }
- public abstract class Car implements Insurance
  {
  public abstract void insuranceDescription (String s) { }
  }
- public abstract class Car implements Insurance { } ✔

Question 9

Correct

Mark 1.00 out of 1.00

Flag question

Predict the output.

```
interface DoStuff2
{
        float getRange(int low, int high);
}
interface DoMore
{
        float getAvg(int a, int b, int c);
}
abstract class DoAbstract implements DoStuff2, DoMore
{ }
class DoStuff implements DoStuff2
{
        public float getRange(int x, int y)
        {
                return 3.14f;
        }
}
interface DoAll extends DoMore
{
        float getAvg(int a, int b, int c, int d);
}
```

Select one:

- The file will compile without error. ✔
- Compile time Error
- Runtime Error

Question 10

Correct

Mark 1.00 out of 1.00

Flag question

implements | extends

```
interface ICalculate
{
  public int add(int a,int b);
}
interface IScientificCalculator [extends] ✔ Calculator
{
  public int calcSine(double value);
}
```

Question 11

Correct

Mark 1.00 out of 1.00

Flag question

extends | implements

```
interface ICalculate
{
  public int add(int a,int b);
}

class Test [implements] ✔ ICalculate
{
  public int add(int a,int b)
  {
     //some code
  }
}
```

Question 12

Correct

Mark 4.00 out of 4.00

Flag question

abstract | Bluetooth | Mobile | extends | implements | super | this

Drag and drop the correct option.

```
interface Bluetooth
{
  public void sendFile();
}
abstract class Mobile
{
  abstract public void makeCall();
}
class IPhone [extends] [Mobile] [implements] [Bluetooth]
{

   //some code
}
```

Question 13

Correct

Mark 1.00 out of 1.00

Flag question

Which of the following are valid statements about interfaces?

Select one or more:

- [x] All methods that are declared in an interface are by default public and abstract. ✔
- [ ] Interface can contain only one static method with definition provided.
- [ ] Interface cannot contain variables.
- [x] Interface can contain any number of default methods which needs to be provided with implementation. ✔

Question 14

Correct

Mark 1.00 out of 1.00

Flag question

```
interface Loan
{
    double issueLoan(double salary);
    double repayLoan(double loanAmt,double amount);
}

class Employee implements Loan
{
    double issueLoan(double salary) {
        System.out.println("Issue Loan");
        return salary * 0.10;
    }
    double repayLoan(double loanAmt,double amount) {
        System.out.println("Repay Loan");
        return loanAmt - amount;
    }
}
```

What is the output of the above code when executed as :

```
Employee e = new Employee();
e.issueLoan(500000);
```

Select one:

- ( ) The function returns 50000
- ( ) Runtime exception
- (•) Compilation error ✔
- ( ) Interface Loan can have only one abstract method