ME-Q1) Given the following definition of the classes Animal, Lion, and Jumpable, select the correct combinations of assignments of a variable that don't result in compilation errors or runtime exceptions (select 2 options).

```
interface Jumpable {}
class Animal {}
class Lion extends Animal implements Jumpable {}
```

- **a** Jumpable var1 = newJumpable();
- **b** Animal var2 = newAnimal();
- **c** Lion var3 = newAnimal();
- **d** Jumpable var4 = newAnimal();
- **e** Jumpable var5 = newLion();
- **f** Jumpable var6 = (Jumpable)(newAnimal());

ME-Q2) Given the following code, which option, if used to replace /*INSERT CODE HERE */, will make the code print 1? (Select 1 option.)

```
try {
    String[][] names = {{"Andre", "Mike"}, null, {"Pedro"}};
    System.out.println (names[2][1].substring(0, 2));
} catch (/*INSERT CODE HERE*/) {
    System.out.println(1);
}
```

- **a** IndexPositionException e
- **b** NullPointerException e
- **c** ArrayIndexOutOfBoundsException e
- **d** ArrayOutOfBoundsException e

ME-Q3) What is the output of the following code? (Select 1 option.)

```
public static void main(String[] args) { int a
    = 10; String name = null;
    try {
        a = name.length();                    //line1
        a++;                                  //line2
    } catch (NullPointerException e){
        ++a;
        return;
    } catch (RuntimeException e){
        a--;
```

```
            return;
        } finally {
            System.out.println(a);
        }
}
```

- **a** 5
- **b** 6
- **c** 10
- **d** 11
- **e** 12
- **f** Compilation error
- **g** No output
- **h** Runtime exception

ME-Q4) Given the following class definition,

class Student { int marks = 10; }

what is the output of the following code? (Select 1 option.)

```
class Result {
    public static void main(String... args) {
        Student s = new Student();
        switch (s.marks) {
            default: System.out.println("100");
            case 10: System.out.println("10");
            case 98: System.out.println("98");
        }
    }
}
```

- **a** 100
  10
  98
- **b** 10
  98
- **c** 100
- **d** 10

ME-Q5) Given the following code, which code can be used to create and initialize an object of the class ColorPencil? (Select 2 options.)

```
class Pencil {}
class ColorPencil extends Pencil {
    String color;
    ColorPencil(String color) {this.color = color;}
}
```

- **a** ColorPencil var1 = new ColorPencil();
- **b** ColorPencilvar2=newColorPencil(RED);
- **c** ColorPencilvar3=newColorPencil("RED");
- **d** Pencil var4=newColorPencil("BLUE");

ME-Q6) What is the output of the following code? (Select 1 option.)

```
class Doctor { protected
    int age;
    protected void setAge(int val) { age = val; }
    protected int getAge() { return age; }
}
class Surgeon extends Doctor {
    Surgeon(String val) {
        specialization = val;
    }
```

```
    String specialization;
    String getSpecialization() { return specialization; }
}
class Hospital {
    public static void main(String args[]) {
        Surgeon s1 = new Surgeon("Liver");
        Surgeon s2 = new Surgeon("Heart");
        s1.age = 45;
        System.out.println(s1.age + s2.getSpecialization());
        System.out.println(s2.age + s1.getSpecialization());
    }
}
```

- **a** 45Heart
  0Liver
- **b** 45Liver
  0Heart
- **c** 45Liver
  45Hear
  t
- **d**
  45Hear
  t
  45Hear
  t
- **e** Class fails tocompile.

ME-Q7) What is the output of the following code? (Select 1 option.)

```
class RocketScience {
    public static void main(String args[]) { int a
        = 0;
        while (a == a++) {
            a++;
            System.out.println(a);
        }
    }
}
```

- **a** The whileloop won't execute; nothing will be printed.
- **b** The whileloop will execute indefinitely, printing all numbers, starting from 1.
- **c** The whileloop will execute indefinitely, printing all even numbers, starting from 0.
- **d** The whileloop will execute indefinitely, printing all even numbers, starting from 2.
- **e** The whileloop will execute indefinitely, printing all odd numbers, starting from 1.
- **f** The whileloop will execute indefinitely, printing all odd numbers, starting from 3.

ME-Q8) Given the following statements,

- com.ejavais a package
- class Personis defined in package com.ejava
- class Courseis defined in package com.ejava

which of the following options correctly import the classes Person and Course in the class MyEJava? (Select 3 options.)

- **a** import com.ejava.*;
  class MyEJava {}
- **b** import com.ejava;
  class MyEJava {}
- **c** import com.ejava.Person;
  import com.ejava.Course;
  class MyEJava {}
- **d** import
  com.ejava.Person;
  import com.ejava.*;
  class MyEJava {}

ME-Q9) Given that the following classes Animaland Forestare defined in the same package, examine the code and select the correct statements (select 2 options).

```
line1>      class Animal {
line2>          public void printKing() { line3>
                    System.out.println("Lion");
line4>          }
line5>      }

line6>      class Forest {
line7>          public static void main(String... args){ line8>
                    Animal anAnimal = new Animal();
line9>              anAnimal.printKing();
```

```
line10>            }
line11>      }
```

- **a** The class Forestprints Lion.
- **b** If the code on line 2 is changed as follows, the class Forestwill print Lion:

  private void printKing() {

- **c** If the code on line 2 is changed as follows, the class Forestwill print Lion:

  void printKing() {

- **d** If the code on line 2 is changed as follows, the class Forestwill print Lion:

  default void printKing() {

ME-Q10) Given the following code,

```
class MainMethod {
    public static void main(String... args) {
        System.out.println(args[0]+":"+ args[2]);
    }
}
```

what is its output if it's executed using the following command? (Select 1 option.)

java MainMethod 1+2 2*3 4-3 5+1

- **a** java:1+2
- **b** java:3
- **c** MainMethod:2*3
- **d** MainMethod:6
- **e** 1+2:2*3
- **f** 3:3
- **g** 6
- **h** 1+2:4-3
- **i** 31
- **j** 4

ME-Q11) What is the output of the following code? (Select 1 option.)

```
interface Moveable {
      int move(int distance);
}
class Person {
     static int MIN_DISTANCE = 5;
     int age;
     float height;
     boolean result;
     String name;
}
public class EJava {
     public static void main(String arguments[]) {
          Person person = new Person();
          Moveable moveable = (x) -> Person.MIN_DISTANCE + x;
          System.out.println(person.name + person.height + person.result
                                                + person.age + moveable.move(20));
     }
}
```

- **a** null0.0false025
- **b** null0false025
- **c** null0.0ffalse025
- **d** 0.0false025
- **e** 0false025
- **f** 0.0ffalse025
- **g** null0.0true025
- **h** 0true025
- **i** 0.0ftrue025
- **j** Compilationerror
- **k** Runtime exception

ME-Q12) Given the following code, which option, if used to replace /*INSERTCODE HERE */,willmakethecodeprintthevalueofthevariablepagesPerMin?(Select1 option.)

```
class Printer { int
     inkLevel;
}
class LaserPrinter extends Printer { int
```

```
    pagesPerMin;
    public static void main(String args[]) { Printer
        myPrinter = new LaserPrinter();
        System.out.println(/* INSERT CODE HERE*/);
    }
}
```

- **a** (LaserPrinter)myPrinter.pagesPerMin
- **b** myPrinter.pagesPerMin
- **c** LaserPrinter.myPrinter.pagesPerMin
- **d** ((LaserPrinter)myPrinter).pagesPerMin

ME-Q13) What is the output of the following code? (Select 1 option.)

```
interface Keys {
    String keypad(String region, int keys);
}
public class Handset {
    public static void main(String... args){ double
        price;
        String model;
        Keys varKeys = (region, keys) ->
                        {if (keys >= 32)
                          return region; else return "default";};
        System.out.println(model + price + varKeys.keypad("AB", 32));
    }
}
```

- **a** null0AB
- **b** null0.0AB
- **c** null0default
- **d** null0.0default
- **e** 0
- **f** 0.0
- **g** Compilation error

ME-Q14) What is the output of the following code? (Select 1 option.)

```
public class Sales {
    public static void main(String args[]) { int
        salesPhone = 1;
        System.out.println(salesPhone++ + ++salesPhone +
                                                    ++salesPhone);
    }
}
```

- **a** 5
- **b** 6
- **c** 8
- **d** 9

ME-Q15) Which of the following options defines the correct structure of a Java class that compiles successfully? (Select 1 option.)

- **a**
```
package com.ejava.guru;
package com.ejava.oracle;
class MyClass {
    int age = /* 25 */ 74;
}
```
- **b**
```
import com.ejava.guru.*;
import com.ejava.oracle.*;
package com.ejava;
class MyClass {
    String name = "e" + "Ja /*va*/ v";
}
```
- **c**
```
class MyClass{
    import com.ejava.guru.*;
}
```
- **d**
```
class MyClass{
    int abc;
    String course = //this is a comment
                "eJava";
}
```
- **e** None of the above

ME-Q16) What is the output of the following code? (Select 1 option.)

```
class OpPre {
    public static void main(String... args){ int x =
        10;
        int y = 20;
        int z = 30;
          if (x+y%z > (x+(-y)*(-z))) {
              System.out.println(x + y + z);
          }
    }
}
```

- **a** 60
- **b** 59
- **c** 61
- **d** No output.
- **e** The code fails to compile.

ME-Q17) Select the most appropriate definition of the variable name and the line num- ber on which it should be declared so that the following code compiles successfully (choose 1 option).

```
class EJava {
    // LINE 1
    public EJava() {
        System.out.println(name);
    }
    void calc() {
        // LINE 2
        if (8 > 2) {
            System.out.println(name);
        }
    }
    public static void main(String... args) {
        // LINE 3
        System.out.println(name);
    }
}
```

- **a** Define static String name;on line 1.
- **b** DefineStringname;online1.
- **c** DefineStringname;online2.
- **d** DefineStringname;online3.

ME-Q18) Examine the following code and select the correct statement (choose 1 option).

```
line1>      class Emp {
line2>          Emp mgr = new
Emp(); line3>  }
line4>      class Office {
line5>          public static void main(String args[]){
line6>              Emp e = null;
line7>              e = new Emp();
line8>              e = null;
line9>          }
line10>     }
```

- **a** Theobjectreferredtobyobjecteiseligibleforgarbagecollectiononline8.
- **b** Theobjectreferredtobyobjecteiseligibleforgarbagecollectiononline9.
- **c** The object referred to by object eisn't eligible for garbage collection because its member variable mgrisn't set to null.
- **d** The code throws a runtime exception and the code execution never reaches line 8 or line 9.

ME-Q19) Given the following,

long result;

which options are correct declarations of methods that accept two String arguments and an int argument and whose return value can be assigned to the variable result? (Select 3 options.)

- **a** ShortmyMethod1(Stringstr1,intstr2,Stringstr3)
- **b** Int myMethod2(Stringval1,intval2,Stringval3)
- **c** Byte myMethod3(Stringstr1,str2,inta)
- **d** Float myMethod4(String val1, val2, int val3)
- **e** Long myMethod5(int str2, String str3, String str1)
- **f** Long myMethod6(String... val1, intval2)
- **g** Short myMethod7(int val1, String...val2)

ME-Q20) Which of the following will compile successfully? (Select 3 options.)

- **a** int eArr1[] = {10, 23, 10, 2};
- **b** int[] eArr2 = newint[10];
- **c** int[] eArr3 = new int[]{};
- **d** int[] eArr4 = new int[10]{};
- **e** int eArr5[] = new int[2] {10, 20};

ME-Q21) Assume that Oracle has asked you to create a method that returns the concatenated value of two String objects. Which of the following methods can accom- plish this job? (Select 2 options.)

- **a**
```
public String add(String 1, String 2) { return
        str1 + str2;
    }
```
- **b**
```
private String add(String s1, String s2) { return
        s1.concat(s2);
    }
```
- **c**
```
protected String add(String value1, String value2) {
        return value2.append(value2);
    }
```
- **d**
```
String subtract(String first, String second) { return
        first.concat(second.substring(0));
    }
```

ME-Q22) Given the following,

```
int ctr = 10;
char[] arrC1 = new char[]{'P','a','u','l'};
char[] arrC2 = {'H','a','r','r','y'};
//INSERT CODE HERE
System.out.println(ctr);
```

which options, when inserted at //INSERTCODEHERE, will output 14? (Choose 2 options.)

a
```
for (char c1 : arrC1){
        for (char c2 : arrC2) { if (c2
            == 'a') break;
            ++ctr;
        }
    }
```

b
```
for (char c1 :arrC1)
        for (char c2 : arrC2) { if (c2
            == 'a') break;
            ++ctr;
        }
```

c
```
for (char c1 :arrC1)
        for (char c2 : arrC2)
            if (c2 == 'a') break;
            ++ctr;
```

d
```
for (char c1 : arrC1){
        for (char c2 : arrC2) {
            if (c2 == 'a') continue;
            ++ctr;
        }
    }
```

ME-Q23) Given the following definitions of the class ChemistryBook, select the statements that are correct individually (choose 2 options).

```
import java.util.ArrayList;
class ChemistryBook {
    public void read() {}                          //METHOD1
    public String read() { return null; }
                                                   //METHOD2
    ArrayList read(int a) { return null; }
                                                   //METHOD3
}
```

- **a** Methods marked with //METHOD1and //METHOD2are correctly overloaded methods.
- **b** Methods marked with //METHOD2and //METHOD3are correctly overloaded methods.
- **c** Methods marked with //METHOD1and //METHOD3are correctly overloaded methods.
- **d** All the methods—methods marked with //METHOD1, //METHOD2, and //METHOD3—are correctly overloaded methods.

ME-Q24) Given the following,

```
final class Home {
    String name;
    int rooms;
    //INSERT CONSTRUCTOR HERE
}
```

which options, when inserted at //INSERT CONSTRUCTOR HERE, will define valid overloaded constructors for the class Home? (Choose 3 options.)

- **a** Home() {}
- **b** Float Home() {}
- **c** protected Home(int rooms){}
- **d** final Home() {}
- **e** private Home(long name){}
- **f** float Home(int rooms, String name){}
- **g** static Home() {}

ME-Q25) Given the following code, which option, if used to replace //INSERTCODE HERE, will make the code print numbers that are completely divisible by 14? (Select 1 option.)

```
for (int ctr = 2; ctr <= 30; ++ctr) { if (ctr
    % 7 != 0)
        //INSERT CODE HERE
    if (ctr % 14 == 0)
        System.out.println(ctr);
}
```

- **a** continue;
- **b** exit;
- **c** break;
- **d** end;

ME-Q26) What is the output of the following code? (Select 1 option.)

```
import java.util.function.Predicate;
public class MyCalendar {
    public static void main(String arguments[]) {
        Season season1 = new Season();
        season1.name = "Spring";

          Season season2 = new Season();
          season2.name = "Autumn";
          Predicate<String> aSeason = (s) -> s == "Summer" ?
                                        season1.name : season2.name;
          season1 = season2;
          System.out.println(season1.name);
          System.out.println(season2.name);
          System.out.println(aSeason.test(new String("Summer")));
    }
}
class Season {
    String name;
}
```

- **a** String
  Autu
  mn
  false
- **b** Spring
  String
  false
- **c** Autumn
  Autu
  mn
  false
- **d**
  Autu
  mn
  String
  true
- **e** Compilation error
- **f** Runtime exception

ME-Q27) What is true about the following code? (Select 1 option.)

```
class Shoe {}
class Boot extends Shoe {}
class ShoeFactory {
    ShoeFactory(Boot val) {
        System.out.println("boot");
    }
    ShoeFactory(Shoe val) {
        System.out.println("shoe");
    }
}
```

- **a** The class ShoeFactory has a total of two overloaded constructors.
- **b** The class ShoeFactory has three overloaded constructors, two user-defined constructors, and one default constructor.
- **c** The class ShoeFactory will fail to compile.
- **d** The addition of the following constructor will increment the number of constructors of the class ShoeFactory to 3:

```
private ShoeFactory (Shoe arg) {}
```

ME-Q28) Given the following definitions of the classes ColorPencil and TestColor, which option, if used to replace //INSERTCODEHERE, will initialize the instance vari- able colorof the reference variable myPencilwith the Stringliteral value "RED"? (Select 1option.)

```
class ColorPencil {
    String color;
      ColorPencil(String color) {
        //INSERT CODE HERE
    }
}
class TestColor {
    ColorPencil myPencil = new ColorPencil("RED");
}
```

- **a** this.color = color;
- **b** color = color;
- **c** color = RED;
- **d** this.color = RED;

ME-Q29) What is the output of the following code? (Select 1 option.)

```
class EJavaCourse {
    String courseName = "Java";
}
class University {
    public static void main(String args[]) {
        EJavaCourse courses[] = { new EJavaCourse(), new EJavaCourse()};
        courses[0].courseName = "OCA";
        for (EJavaCourse c : courses) c = new EJavaCourse();
        for (EJavaCourse c : courses) System.out.println(c.courseName);
    }
}
```

- **a** Java
  Java
- **b** OCA
  Java
- **c**
  OC
  A
  OC
  A
- **d** None of the above

ME-Q30) What is the output of the following code? (Select 1 option.)

```
class Phone {
    static void call() {
        System.out.println("Call-Phone");
    }
}

class SmartPhone extends
    Phone{ static void call() {
        System.out.println("Call-SmartPhone");
    }
}
class TestPhones {
    public static void main(String... args) {
        Phone phone = new Phone();
        Phone smartPhone = new
        SmartPhone(); phone.call();
        smartPhone.call();
    }
}
```

- **a** Call-Phone
  Call-Phone
- **b** Call-Phone Call-
  SmartPhone
- **c** Call-Phone
  null
- **d** null
  Call-SmartPhone

ME-Q31) Given the following code, which of the following statements are true? (Select 3 options.)

```
class MyExam {
    void question() {
        try {
            question();
        } catch (StackOverflowError e) {
            System.out.println("caught");
        }
    }
    public static void main(String args[]) { new
        MyExam().question();
    }
}
```

- **a** The code will print caught.
- **b** The code won't print caught.
- **c** The code would print caught if StackOverflowError were a runtime exception. **d** The code would print caught if StackOverflowError were a checked exception. **e** The code would print caught if question() throws the exception NullPointer-Exception.

ME-Q32) A class Student is defined as follows:

```
public class Student {
    private String fName;
    private String lName;

    public Student(String first, String last) { fName
        = first; lName = last;
    }
    public String getName() { return fName + lName; }
}
```

The creator of the class later changes the method getName as follows:

```
public String getName() { return
    fName + " " + lName;
}
```

What are the implications of this change? (Select 2 options.)

- **a** The classes that were using the class Studentwill fail to compile.
- **b** Theclassesthatwereusingtheclass Studentwillworkwithoutanycompila- tion issues.
- **c** The class Studentis an example of a well-encapsulated class.
- **d** The class Studentexposes its instance variable outside the class.

ME-Q33) What is the output of the following code? (Select 1 option.)

```
class ColorPack {
    int shadeCount = 12;
    static int getShadeCount() {
        return shadeCount;
    }
}
class Artist {
    public static void main(String args[]) { ColorPack
        pack1 = new ColorPack();
        System.out.println(pack1.getShadeCount());
    }
}
```

- **a** 10
- **b** 12
- **c** No output
- **d** Compilation error

ME-Q34) PauldefinedhisLaptopandWorkshopclassestoupgradehislaptop'smem- ory. Do youthinkhesucceeded? Whatistheoutputofthiscode? (Select1option.)

```
class Laptop {
    String memory = "1 GB";
}
class Workshop {
    public static void main(String args[]) {
        Laptop life = new Laptop(); repair(life);

            System.out.println(life.memory);
      }
      public static void repair(Laptop laptop) { laptop.memory
            = "2 GB";
      }
  }
```

- **a** 1GB
- **b** 2GB
- **c** Compilation error
- **d** Runtime exception

ME-Q35) What is the output of the following code? (Select 1 option.)

```
public class Application {
    public static void main(String... args){ double
        price = 10;
        String model; if
        (price > 10)
            model =
        "Smartphone"; else if
        (price <= 10)
            model = "landline";
        System.out.println(model);
    }
}
```

- **a** landline
- **b** Smartphone
- **c** Nooutput
- **d** Compilation error

ME-Q36) What is the output of the following code? (Select 1 option.)

```
class EString {
    public static void main(String args[]) {
        String eVal = "123456789";
        System.out.println(eVal.substring(eVal.indexOf("2"),
    ➥ eVal.indexOf("0")).concat("0"));
    }
}
```

- **a** 234567890
- **b** 34567890
- **c** 234456789
- **d** 3456789
- **e** Compilation error
- **f** Runtime exception

ME-Q37) Examine the following code and select the correct statements (choose 2 options).

```
class Artist  { Artist
      assistant;
}
class Studio {
      public static void main(String... args) { Artist
            a1 = new Artist();
            Artist a2 = new Artist();
            a2.assistant = a1;
            a2 = null;                // Line 1
      }
       // Line 2
}
```

- **a** At least two objects are garbage collected on line 1.
- **b** At least one object is garbage collected on line 1.
- **c** No objects are garbage collected on line 1.
- **d** The number of objects that are garbage collected on line 1 is unknown.
- **e** At least two objects are eligible for garbage collection on line 2.

ME-Q38) What is the output of the following code? (Select 1 option.)

```
class Book {
      String ISBN;
      Book(String val) {
            ISBN = val;
      }
}
class TestEquals {
      public static void main(String... args){ Book
            b1 = new Book("1234-4657");  Book b2
            = new Book("1234-4657");
            System.out.print(b1.equals(b2) +":");
            System.out.print(b1 == b2);
      }
}
```

- **a** true:false
- **b** true:true
- **c** false:true
- **d** false:false
- **e** Compilation error—there is no equalsmethod in the class Book.
- **f** Runtime exception.

ME-Q39) Which of the following statements are correct? (Select 2 options.)

- **a** StringBuilder sb1 = new StringBuilder() will create a StringBuilder object with no characters but with an initial capacity to store 16 characters.
- **b** StringBuildersb1=newStringBuilder(5*10)will create a StringBuilder object with a value of 50.
- **c** Unlike the class String, the concatmethod in StringBuildermodifies the value of a StringBuilderobject.
- **d** The insertmethod can be used to insert a character, number, or Stringat the start or end or a specified position of a StringBuilder.

ME-Q40) Given the following definition of the class Animaland the interface Jump, select the correct array declarations and initialization (choose 3 options).

```
interface Jump {}
class Animal implements Jump {}
```

- **a** Jump eJump1[] = {null, new Animal()};
- **b** Jump[] eJump2 = newAnimal()[22];
- **c** Jump[]eJump3=newJump[10];
- **d** Jump[]eJump4=newAnimal[87];
- **e** Jump[]eJump5=newJump()[12];

ME-Q41) What is the output of the following code? (Select 1 option.)

```
import java.util.*;
class EJGArrayL {
    public static void main(String args[]) { ArrayList<String>
        seasons = new ArrayList<>(); seasons.add(1, "Spring");
        seasons.add(2, "Summer"); seasons.add(3, "Autumn");
        seasons.add(4, "Winter"); seasons.remove(2);

        for (String s : seasons)
            System.out.print(s + ", ");
    }
}
```

a Spring, Summer, Winter,

b Spring, Autumn, Winter,

c Autumn, Winter,

d Compilation error

e Runtime exception

ME-Q42) What is the output of the following code? (Select 1 option.)

```
class EIf {
    public static void main(String args[]) { bool
        boolean = false;
        do {
            if (boolean = true)
                System.out.println("true");
            else
                System.out.println("false");
        }
        while(3.3 + 4.7 > 8);          }
}
```

- **a** The class will print true.
- **b** The class will print false.
- **c** The class will print true if the if condition is changed to boolean==true.
- **d** The class will print false if the if condition is changed to boolean!=true.
- **e** The class won't compile.
- **f** Runtime exception.

ME-Q43) How many Fish did the Whale (defined as follows) manage to eat? Examine the following code and select the correct statements (choose 2 options).

```
class Whale {
    public static void main(String args[]) {
        boolean hungry = false;
         while (hungry=true) {
             ++Fish.count;
        }
        System.out.println(Fish.count);
    }
}
class Fish {
    static byte count;
}
```

- **a** The code doesn't compile.
- **b** The code doesn't print a value.
- **c** The code prints 0.
- **d** Changing ++Fish.count to Fish.count++ will give the same results.

ME-Q44) Given the following code, which option, if used to replace /*REPLACECODE HERE */, will make the code print the name of the phone with the position at which it's stored in the array phones? (Select 1 option.)

```
class Phones {
    public static void main(String args[]) {
        String phones[]= {"BlackBerry", "Android", "iPhone"};

          for (String phone : phones)
              /* REPLACE CODE HERE */
     }
}
```

- **a** System.out.println(phones.count + ":" +phone);
- **b** System.out.println(phones.counter + ":" +phone);
- **c** System.out.println(phones.getPosition()+":"+phone);
- **d** System.out.println(phones.getCtr()+":"+phone);
- **e** System.out.println(phones.getCount()+":"+phone);
- **f** System.out.println(phones.pos+":"+phone);
- **g** None of the above

ME-Q45) Given the following code,

```
Byte b1 = (byte)100;                          // 1
Integer i1 = (int)200;                        // 2
Long l1 = (long)300;                          // 3
Float f1 = (float)b1 + (
      0int)l1;                  // 4
String s1 = 300;                              // 5
if (s1 == (b1 + i1))                          // 6
     s1 = (String)500;                        // 7
else                                          // 8
     f1 = (int)100;                           // 9
System.out.println(s1 + ":" + f1);            // 10
```

what is the output? Select 1 option.

- **a** Code fails compilation at line numbers 1, 3, 4, 7.
- **b** Code fails compilation at line numbers 6, 7.
- **c** Code fails compilation at line numbers 7, 9.
- **d** Code fails compilation at line numbers 4, 5, 6, 7, 9.
- **e** No compilation error—outputs 500:300.
- **f** No compilation error—outputs 300:100.
- **g** Runtime exception.

ME-Q46) What is the output of the following code? (Select 1 option.)

```
class Book {
      String ISBN;
      Book(String val) {
            ISBN = val;
      }
      public boolean equals(Object b) { if
            (b instanceof Book) {
                  return ((Book)b).ISBN.equals(ISBN);
            }


          else
               return false;

     }
}

class TestEquals {
     public static void main(String args[]){ Book
          b1 = new Book("1234-4657"); Book b2
          = new Book("1234-4657"); LocalDate
          release = null;
          release = b1.equals(b2) ? b1 == b2? LocalDate.of(2050,12,12):
          LocalDate.parse("2072-02-01"):LocalDate.parse("9999-09-09");
          System.out.print(release);
     }
}
```

- **a** 2050-12-12
- **b** 2072-02-01
- **c** 9999-09-09
- **d** Compilation error
- **e** Runtime exception

ME-Q47) What is the output of the following code? (Select 1 option.)

```
int a = 10;
for (; a <= 20; ++a) {
    if (a%3 == 0) a++; else if (a%2 == 0) a=a*2;
    System.out.println(a);
}
```

- **a** 11
  13
  15
  17
  19
- **b** 20
- **c** 11
  14
  17
  20
- **d** 40
- **e** Compilation error

ME-Q48) Given the following code, which option, if used to replace //INSERTCODE HERE, will define an overloaded rideWavemethod? (Select 1 option.)

```
class Raft {
    public String rideWave() { return null; }
    //INSERT CODE HERE
}
```

- **a** public String[] rideWave() { return null; }
- **b** protected void riceWave(int a){}
- **c** private void rideWave(int value, String value2) {}
- **d** defaultStringBuilderrideWave(StringBuffera){returnnull;}

ME-Q49) Given the following code, which option, if used to replace //INSERTCODE HERE, will correctly calculate the sum of all the even numbers in the array numand store it in the variable sum? (Select 1 option.)

```
int num[] = {10, 15, 2, 17};
int sum = 0;
for (int number : num) {
    //INSERT CODE HERE
     sum += number;
}
```

- **a** if (number % 2 == 0)
  continue;
- **b** if (number % 2 == 0)
  break;
- **c** if (number % 2 != 0)
  continue;
- **d** if (number % 2 != 0)
  break;

ME-Q50) What is the output of the following code? (Select 1 option.)

```
class Op {
    public static void main(String... args) { int a
        = 0;
        int b = 100;
        Predicate<Integer> compare = (var) -> var++ == 10; if
        (!b++ > 100 && compare.test(a)) {
            System.out.println(a+b);
        }
    }
}
```

- **a** 100
- **b** 101
- **c** 102
- **d** Code fails to compile.
- **e** No output is produced.

ME-Q51) Choose the option that meets the following specification: Create a well-encapsulated class Pencil with one instance variable model. The value of model should be accessible and modifiable outside Pencil. (Select 1 option.)

- **a**
```
class Pencil{
    public String model;
}
```
- **b**
```
class Pencil{
    public String model;
    public String getModel() { return model; }
    public void setModel(String val) { model = val; }
}
```
- **c**
```
class Pencil{
    private String model;
    public String getModel() { return model; }
    public void setModel(String val) { model = val; }
}
```
- **d**
```
class Pencil{
    public String model;
    private String getModel() { return model; } private
    void setModel(String val) { model = val;}
}
```

ME-Q52) What is the output of the following code? (Select 1 option.)

```
class Phone {
    void call() {
        System.out.println("Call-Phone");
    }
}
class SmartPhone extends
    Phone{ void call() {
        System.out.println("Call-SmartPhone");
    }
}
class TestPhones {
    public static void main(String[] args) {
        Phone phone = new Phone();
        Phone smartPhone = new
        SmartPhone(); phone.call();
        smartPhone.call();
    }
}
```

- **a** Call-Phone
  Call-Phone
- **b** Call-Phone Call-
  SmartPhone
- **c** Call-Phone
  null
- **d** null
  Call-SmartPhone

ME-Q53) What is the output of the following code? (Select 1 option.)

```
class Phone {
    String keyboard = "in-built";
}
class Tablet extends Phone {
    boolean playMovie = false;
}
class College2 {
    public static void main(String args[]) {
        Phone phone = new Tablet();
        System.out.println(phone.keyboard + ":" + phone.playMovie);
    }
}
```

- **a** in-built:false
- **b** in-built:true
- **c** null:false
- **d** null:true
- **e** Compilation error

ME-Q54) What is the output of the following code? (Select 1 option.)

```
public class Wall {
    public static void main(String args[]) {
        double area = 10.98;
        String color; if
        (area < 5)
            color = "red";
        else
            color = "blue";
        System.out.println(color);
    }
}
```

- **a** red
- **b** blue
- **c** No output
- **d** Compilation error

ME-Q55) What is the output of the following code? (Select 1 option.)

```
class Diary {
    int pageCount = 100;
    int getPageCount() {
        return pageCount;
    }
    void setPageCount(int val) {
        pageCount = val;
    }
}

class ClassRoom {
    public static void main(String args[]) {
        System.out.println(new Diary().getPageCount());
        new Diary().setPageCount(200);
        System.out.println(new Diary().getPageCount());
    }
}
```

- **a** 100
  200
- **b** 100
  100
- **c** 200
  200
- **d** Code fails to compile.

ME-Q56) How many times do you think you can shop with the following code (that is, what's the output of the following code)? (Select 1 option.)

```
class Shopping {
    public static void main(String args[]) {
        boolean bankrupt = true;
        do System.out.println("enjoying shopping"); bankrupt = false;
        while (!bankrupt);
    }
}
```

- **a** The code prints enjoyingshoppingonce.
- **b** The code prints enjoyingshoppingtwice.
- **c** The code prints enjoyingshoppingin an infinite loop.
- **d** The code fails to compile.

ME-Q57) Which of the following options are valid for defining multidimensional arrays? (Choose 4options.)

- **a** String ejg1[][] = newString[1][2];
- **b** String ejg2[][] = new String[][] { {}, {} };
- **c** String ejg3[][] = newString[2][2];
- **d** Stringejg4[][]=newString[][]{{null},newString[]{"a","b","c"}, {new String()}};
- **e** String ejg5[][] = newString[][2];
- **f** String ejg6[][] = new String[][]{"A", "B"};
- **g** String ejg7[][] = new String[]{{"A"}, {"B"}};

ME-Q58) What is the output of the following code? (Select 1 option.)

```
class Laptop {
    String memory = "1GB";
}
class Workshop {
    public static void
        main(String args[]) {
        Laptop life = new
        Laptop(); repair(life);
        System.out.println(li
        fe.memory);
    }
    public static void repair(Laptop
        laptop){ laptop = new
        Laptop();
        laptop.memory = "2GB";
    }
}
```

- **a** 1GB
- **b** 2GB
- **c** Compilation error
- **d** Runtime exception

ME-Q59) Given the following code, which option, if used to replace //INSERT CODE HERE, will enable a reference variable of type Roamable to refer to an object of the Phone class? (Select 1 option.)

```
int
erf
ace
Roa
ma
ble{
}
clas
s
Pho
ne
{}
class Tablet extends Phone implements Roamable {
    //INSERT CODE HERE
}
```

- **a** Roamable var = newPhone();
- **b** Roamable var =(Roamable)Phone();

- **c** Roamable var = (Roamable)newPhone();
- **d** Because the interface Roamableand the class Phoneare unrelated, a refer- encevariableoftypeRoamablecan'trefer toanobjectoftheclassPhone.

ME-Q60) What is the output of the following code? (Select 1 option.)

```
class Paper {
      Paper() {
            this(10);
            System.out.pri
            ntln("Paper:0"
            );
      }
      Paper(int a) { System.out.println("Paper:1"); }
}
class PostIt extends Paper {}


class TestPostIt {
    public static void
          main(String[] args) {
          Paper paper = new
          PostIt();
    }
}
```

- **a** Paper:1
- **b** Paper:0
- **c**

  P
  a
  p
  e
  r
  :
  0

  P

a
p
e
r
:
1

d

P
a
p
e
r
:
1

P
a
p
e
r
:
0

ME-Q61) Examine the following code and select the correct statement (choose 1 option).

```
line1> class StringBuilders {
line2>        public static void main(String... args) {
line3>              StringBuilder sb1 = new
StringBuilder("eLion"); line4>         String ejg =
null;
line5>              ejg =
      sb1.append("X").substring(sb1.indexOf("L"),
      sb1.indexOf("X"));
line6>
              System.out.
println(ejg); line7>  }
line8> }
```

a The code will print LionX.

- **b** The code will print Lion.
- **c** The code will print Lionif line 5 is changed to the following:

  ejg = sb1.append("X").substring(sb1.indexOf('L'), sb1.indexOf('X'));

- **d** The code will compile only when line 4 is changed to the following:

  StringBuilder ejg = null;

ME-Q62) Given the following code,

```
int
    e
    r
    f
    a
    c
    e

    J
    u
    m
    p
    a
    b
    l
    e

    {

    i
    n
    t

    h
    e
    i
    g
    h
    t

    =

    1
    ;
    default void
        worldRe
        cord() {
```

```
            System.o
            ut.print(
            height);
        }
}
int
    e
    r
    f
    a
    c
    e

    M
    o
    v
    e
    a
    b
    l
    e

    {

    i
    n
    t

    h
    e
    i
    g
    h
    t

    =

    2
    ;
    static void
            worldRe
            cord() {
            System.o
            ut.print(
            height);
        }
}
```

```
class Chair implements
    Jumpable, Moveable {
    int height = 3;
    Chair() {
        worldRecord();
    }
    public static void
        main(String args[]) {
        Jumpable j = new
        Chair();
        Moveabl
        e m =
        new
        Chair();
        Chair c =
        new
        Chair();
    }
}
```

what is the output? Select 1 option.

- **a** 111
- **b** 123
- **c** 333
- **d** 222
- **e** Compilation error
- **f** Runtime exception

ME-Q63) Given the following code, which option, if used to replace /* INSERT CODE HERE */, will enable the class Jungle to determine whether the reference variable animal refers to an object of the class Lion and print 1? (Select 1 option.)

```
class Animal{ float age; }
class Lion extends
Animal { int claws;}
class Jungle {
    public static void
        main(String args[]) {
        Animal animal = new
        Lion();
        /* INSERT CODE HERE */
            System.out.println(1);
```

```
    }
}
```

- **a** if (animal instanceof Lion)
- **b** if (animal instanceOf Lion)
- **c** if (animal == Lion)
- **d** if (animal =Lion)

ME-Q64) Given that the file Test.java, which defines the following code, fails to com- pile, select the reasons for the compilation failure (choose 2 options).

```
class Perso
n {
Perso
n(Strin
g
value)
{}
}
class Employee extends Person {}

class Test {
    public static void
        main(String args[]) {
        Employee e = new
        Employee();
    }
}
```

- **a** The class Personfails to compile.
- **b** The class Employeefails to compile.
- **c** The default constructor can call only a no-argument constructor of a base class.
- **d** Thecodethatcreatestheobjectoftheclass Employeeinthe classTestdid notpassaStringvaluetotheconstructorofthe class Employee.

ME-Q65) Examine the following code and select the correct statements (choose 2 options).

```
class Bottle {
      void Bottle() {}
      void Bottle(WaterBottle w) {}
}
class WaterBottle extends Bottle {}
```

- **a** A base class can't pass reference variables of its defined class as method parameters in constructors.
- **b** The class compiles successfully—a base class can use reference variables of its derived class as method parameters.
- **c** The class Bottledefines two overloaded constructors.
- **d** The class Bottlecan access only one constructor.

ME-Q66) Given the following code, which option, if used to replace /* INSERTCODE HERE*/, will cause the code to print 110? (Select 1 option.)

```
class Book {
      private int pages = 100;
}
class Magazine
      extends
      Book {
      private int
      interviews
      = 2;
      private int totalPages() { /* INSERT CODE HERE */ }

      public static void main(String[] args) {
            System.out.println(new
            Magazine().totalPages());
      }

}
```

- **a** returnsuper.pages+ this.interviews*5;
- **b** returnthis.pages+ this.interviews*5;
- **c** returnsuper.pages+ interviews*5;

- **d** return pages +this.interviews*5;
- **e** None of the above

ME-Q67) Given the following code,

```
class NoInkException
extends Exception {} class
Pen{
    void write(String val) throws
        NoInkException { int c = (10
        - 7)/ (8 - 2 - 6);
    }
    void article() {
        //INSERT CODE HERE
    }
}
```

which of the options, when inserted at //INSERTCODEHERE, will define a valid use of the method write in the method article? (Select 2 options.)

- **a**
```
try{
        new Pen().write("story");
    } catch (NoInkException e) {}
```
- **b**
```
try{
        new Pen().write("story");
    } finally {}
```
- **c**
```
try{
        write("story");
    } catch (Exception e) {}
```
- **d**
```
try{
        new Pen().write("story");
    } catch (RuntimeException e) {}
```

ME-Q68) What is the output of the following code? (Select 1 option.)

```
class EMyMethods {
    static
    String
    name =
    "m1";
    void
```

```
        riverRafti
        ng() {
              St
              rin
              g
              na
              m
              e
              =
              "
              m
              2"
              ; if
              (8
              >
              2)
              {
                    String
                    name =
                    "m3";
                    System.o
                    ut.printl
                    n(name);
              }
        }
        public static void
              main(String[] args) {
              EMyMethods m1 =
              new EMyMethods();
              m1.riverRafting();
        }
  }
```

- **a** m1
- **b** m2
- **c** m3
- **d** The code fails to compile.

ME-Q69) What is the output of the following code? (Select 1 option.)

```
class EBowl {
      public static void
            main(String args[]) {
```

```
        String eFood =
        "Corn";
        System.out.println(e
        Food); mix(eFood);
        System.out.println(e
        Food);
    }
    static void
        mix(String
        foodIn) {
        foodIn.conc
        at("A");
        foodIn.repla
        ce('C', 'B');
    }
}
```

- **a**

  C
  o
  r
  n

  B
  o
  r
  n
  A

- **b**

  C
  o
  r
  n

  C
  o
  r
  n
  A

c

C
o
r
n

B
o
r
n

d

C
o
r
n

C
o
r
n

ME-Q70) Which statement is true for the following code? (Select 1 option.)

```
class SwJava {
    public static void main(String args[]) {
        String[] shapes = {"Circle", "Square",
        "Triangle"}; switch (shapes) {
            case "Square":
            System.out.println("Circle"); break;
            case "Triangle":
            System.out.println("Square"); break;
            case "Circle":
            System.out.println("Triangle"); break;
        }
    }
}
```

a The codeprints Circle.

b The codeprints

Square.

c The codeprints Triangle.

d The codeprints

C
i
r
c
l
e

S
q
u
a
r
e

T
r
i
a
n
g
l
e

e The code prints

T
r
i
a
n
g
l
e

C
i
r
c
l
e

S
q
u
a
r
e

f The code fails to compile.

ME-Q71) Given the following definition of the classes Person, Father, and Home, which option, if used to replace //INSERT CODE HERE, will cause the code to compile successfully? (Select 3 options.)

```
class Person {}
class Father extends Person {
    public void dance() throws ClassCastException {}
}
class Home {
    public static void
        main(String args[]) {
        Person p = new
        Person();
        try {
            ((Father)p).dance();
        }
        //INSERT CODE HERE
    }
}
```

a catch (NullPointerExcepti on e) {} catch (ClassCastException e) {} catch (Exception e) {} catch (Throwable t) {}

b catch (ClassCastExceptio n e) {} catch (NullPointerExcept ion e) {} catch (Exception e) {} catch (Throwable t) {}

c catch (ClassCastExceptio n e) {} catch (Exception e) {} catch (NullPointerExcepti on e) {} catch

(Throwable t) {}

- **d** catch (Throwable t) {} catch (Exception e) {} catch (ClassCastException e) {} catch (NullPointerException e) {}
- **e** finally {}

ME-Q72) What is the output of the following code? (Select 1 option.)

```
import java.time.*;
class Camera {
    public static void main(String args[]) {
        int hours;
        LocalDateTime now =
        LocalDateTime.of(2020, 10, 01, 0 , 0);
        LocalDate before =
        now.toLocalDate().minusDays(1); LocalTime
        after = now.toLocalTime().plusHours(1);

        while (before.isBefore(after) && hours < 4) {
            ++hours;
        }
        System.out.println("Hours:" + hours);
    }
}
```

- **a** The code printsCamera:null.

- **b** The code prints Camera:Adjustsettingsmanually.
- **c** The code printsCamera:.
- **d** The code will fail to compile.

ME-Q73) The output of the class TestEJavaCourse, defined as follows, is 300:

```
class Course {
    int enrollments;
}
class TestEJavaCourse {
    public static void
        main(String args[]) {
        Course c1 = new
        Course();
        Course
        c2 = new
        Course();
        c1.enroll
        ments =
        100;
        c2.enrollments = 200;
        System.out.println(c1.enrollments +
        c2.enrollments);
    }
}
```

Whatwillhappenifthevariableenrollmentsisdefinedasastatic variable?(Select 1 option.)

- **a** No change in output. TestEJavaCourseprints 300.
- **b** Change in output. TestEJavaCourseprints 200.
- **c** Change in output. TestEJavaCourseprints 400.
- **d** The class TestEJavaCourse failstocompile.

ME-Q74) What is the output of the following code? (Select 1 option.)

```
String ejgStr[] = new String[][]{{null},new
    String[]{"a","b","c"},{new String()}}[0] ;
```

```
String
ejgStr1[] =
null;
String
ejgStr2[] =
{null};

System.out.pri
ntln(ejgStr[0]);
System.out.pri
ntln(ejgStr2[0])
;
System.out.pri
ntln(ejgStr1[0])
;
```

- **a** null
  NullP
  ointer
  Excep
  tion
- **b**

  n
  u
  l
  l

  n
  u
  l
  l
  NullPointerException
- **c** NullPointerException
- **d**

  n
  u
  l
  l

  n
  u
  l

l

n
u
l
l

ME-Q75) Examine the following code and select the correct statement (choose 1 option).

```
im
po
rt
ja
va
.ut
il.
*;
cla
ss
Pe
rs
on
{}
class Emp extends Person {}

class TestArrayList {
      public static void main(String[]
            args) { ArrayList<Object> list
            = new ArrayList<>();
            list.add(new String("1234"));                    //LINE1
            list.add(new Person());                          //LINE2
            list.add(new Emp());                             //LINE3
            list.add(new String[]{"abcd", "xyz"});           //LINE4
            list.add(LocalDate.now().plus(1));               //LINE5
      }
}
```

- **a** The code on line 1 won't compile.
- **b** The code on line 2 won't compile.
- **c** The code on line 3 won't compile.
- **d** The code on line 4 won't compile.
- **e** The code on line 5 won't compile.
- **f** None of the above.
- **g** All the options from (a) through (e).

ME-Q76) What is the output of the following code? (Select 1 option.)

```
public class If2 {
    public static void main(String args[]) {
        int a = 10; int b = 20; boolean c = false;
        if (b > a) if (++a == 10) if (c!=true)
        System.out.println(1); else
        System.out.println(2); else
        System.out.println(3);
    }
}
```

- **a** 1
- **b** 2
- **c** 3
- **d** No output

ME-Q77) Given the following code,

```
interface Movable {
    default
        int
        dist
        anc
        e()
        {
        ret
        urn
        10;
    }
}
interface
    Jumpab
    le  {
    default
    int
    distanc
    e(){

       return 10;
   }
}
```

which options correctly define the class Personthat implements interfaces Movable
and Jumpable? (Select 1 option.)

- **a** class Person implements Movable, Jumpable {}
- **b** class Person implements
  Movable, Jumpable {
  default int distance() {
  return 10;
  }
  }
- **c** class Person implements
  Movable, Jumpable {
  public int distance() {
  return 10;
  }
  }
- **d** class Person implements
  Movable, Jumpable {
  public long distance() {
  return 10;
  }
  }
- **e** class Person implements
  Movable, Jumpable { int
  distance() {
  return 10;
  }
  }