

Part I. Single Choice Questions (12×2 points = 24 points)

1. If no any private/protected/public is specified for a member in a class, that member ...
 - A. Is accessible publicly
 - B. Is accessible by the class and its subclasses
 - C. Is only accessible by other classes of the same package
 - D. Is accessible by all classes of the same package
2. The following code

```
int pi = 3.14159;
```

will result in:
 - A. Compilation error
 - B. Runtime exception
 - C. pi being 3.14159
 - D. pi being 3
 - E. pi being 4
3. Assume that we have defined:

```
enum Suit { Spades, Hearts, Diamonds, Clubs };
```

In order to initialize a Suit variable, which of the following is correct?
 - A. Suit s = new Suit(Spades);
 - B. Suit s = Suit.Hearts;
 - C. Suit s = Diamonds;
 - D. Suit s = Suit(Clubs);
4. If class Circle extends class Shape, which of the following code segment is correct?
 - A. Circle x = new Circle(); Object y = (Object)x; Circle z = y;
 - B. Shape y = new Shape(); Circle x = (Circle)y; Shape z = x;
 - C. Circle x = new Circle(); Shape y = x; Circle z = (Circle)y;
 - D. Shape y = new Shape(); Circle x = (Circle)y; Shape z = (Shape)x;
5. What will be the value of num after executing the following statements?

```
int num = 5;  
num += 10.8;  
num = num / 6;
```

 - A. Compilation error
 - B. Runtime exception
 - C. 3
 - D. 2
 - E. 2.5
6. Which one of the following statements is correct?
 - A. "S extends T" is possible when S and T are both interfaces or are both classes
 - B. "S extends T" is only possible when S is an interface and T is a class
 - C. "S extends T" is only possible when S is a class and T is an interface
 - D. "S extends T" is possible with any combination of classes and interfaces
7. What would the following statements display?

```
String word = "12233445566";  
for (int i = 0; i <= 9; i += 3) System.out.print( word.charAt(i));
```

 - A. 1223
 - B. 134
 - C. 1346
 - D. Compilation error
 - E. Runtime exception
8. Late/Dynamic binding is an essential feature for

A. Polymorphism B. Encapsulation C. Serialization D. Modulization

9. Which of the following means that in order for the condition to be true, x is beyond the range [0.0, 1.0) ?

- A. if (x <= 0.0 && x > 1.0)
- B. if (x <= 0.0 || x > 1.0)
- C. if (x < 0.0 || x >= 1.0)
- D. if (x < 0.0 && x >= 1.0)

10. For the class defined as follows:

```
public class Yeh { int x; }
```

Which one of the following code segment cannot compile?

- A. Yeh a = new Yeh(); String s = a.toString();
- B. Yeh a = new Yeh(); boolean b = a.equals("hello");
- C. Yeh a = new Yeh(10);

11. With inheritance (specialization), a subclass object can directly access any

- A. public or private superclass member
- B. private superclass member
- C. public or protected superclass member and none access modifier specified superclass members if it's in the same package
- D. protected, public or private superclass member

12. The following code

```
int a = 0;  
int b = 128 / a;  
System.out.println( b );
```

will result in:

- A. Compilation error: Division must be in a try block
- B. Compilation error: DivideByZero Exception
- C. The program will display 'NaN'
- D. Runtime exception

Part II. Multiple Choices Questions (4×2 points = 8 points)

1. Assume the following definition:

```
[1] public interface Foo {  
[2]     int star = 5;     /* line 2 */  
[3] }
```

Which three modifiers are defined automatically by default in **line 2**?

- A. abstract B. public C. protected D. static E. final

2. We have the class

```

public class Test {
    public float methodOk(float a, float b) { ... }
    // Here
    ...
}

```

Among the following methods, what could we add after the 'Here' comment?

- A. `public int methodOk(int a, int b) { ... }`
 - B. `public int methodOk(float a, float b) { ... }`
 - C. `public float methodOk(float a, float b, float c) { ... }`
 - D. `private float methodOk(int a, int b, int c) { ... }`
 - E. `public int methodOk(String a, String b) { ... }`
3. Which three are valid method signatures in an interface?
- A. `private int getArea();`
 - B. `public float getVol(float x);`
 - C. `public void main(String [] args);`
 - D. `public static void main(String [] args);`
 - E. `boolean setFlag(Boolean [] test);`
4. Which three literals are equal to **int** value **127** in decimal number in Java?
- A. `"127"`
 - B. `127`
 - C. `'127'`
 - D. `0177`
 - E. `0x7F`

Part III. True or False (11×1 point = 11 points)

Determine whether each statement is **true(T)** or **false(F)** in Java according to its correctness.

1. The default statement of a `switch` structure is always executed.
2. A boolean can be compared to an integer.
3. The types of `int` and `double` are not true classes for performance purposes.
4. The `equals` method is an example of a method that is inherited but can be overridden.
5. Every object in Java, no matter what class it came from, has a method called `toString`.
6. Methods that are marked `protected` can be called only in the class and its subclasses.
7. Assume we have an abstract class `T` and an interface `I`. The following declaration is correct:

```
public class X extends T implements I {...}
```
8. An array can store many different types of values.
9. Two strings are compared by the `<`, `<=`, `==`, `=>`, `>` and `!=` operations.
10. An ADT (Abstract Data Type) hides its data structure organization and exposes its operations.
11. In try-with-resources structure, all the resource classes should implements **AutoClosable** interface that has a method **close()** to close the resource.

Part IV: Completion by Matching (10×1.5 point = 15 points)

Choose the **LETTER** that refers to the **BEST ITEM** from the alternative answers listed below to fill in the blanks in each of the following statements.

Alternative Answers:

- | | | |
|----------------|---------------|-----------|
| A. break | G. javac | M. single |
| B. classes | H. JVM | N. static |
| C. compiler | I. new | O. String |
| D. continue | J. null | P. super |
| E. for each | K. Object | Q. this |
| F. interpreter | L. repetition | R. void |

1. The programs that translate high-level language programs into machine language are called _____.
2. Java programs are executed by something called _____.
3. Methods that perform common tasks and do not require creating objects are called _____ methods or class methods.
4. The keyword _____ is used in java to create an object.
5. _____ class has no parent class.
6. All programs can be written in terms of three types of control structures: sequence, selection and _____.
7. Java programmers concentrate on creating _____, which contain fields and the set of methods that manipulate those fields and provide services to clients.
8. The default value of a non-local `int` variable is 0 and the default reference value is _____.

9. The _____ statement allows you to iterate through an array's elements or a Collection without using a counter.
10. The _____ statement, when executed in a repetition statement, skips the remaining statements in the loop body and proceeds with the next iteration of the loop.

Part V. Short Answer Questions (12 points)

1. What does it print after executing the following program? (4 points)

```
import java.util.ArrayList;
public class CodeReading1 {
    public static void main (String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add( "Physics" );
        list.add( "Biology" );
        list.add( "Algebra" );
        list.add( "Programming" );
        list.add( list.indexOf( "Biology" ), "Logic" );
        list.remove( 2 );
        System.out.println( list );
    }
}
```

2. What does it print after executing the following program? (2 points)

```
public class CodeReading2 {
    public CodeReading2 () {
        System.out.print( ":-) " );
    }
    public CodeReading2 (int n) {
        this();
        System.out.println( "Hi " + n );
    }
    public static void main (String[] args) {
        new CodeReading2( 2020 );
    }
}
```

3. What does it print after executing as the following CodeReading3.java? (2 points)

```
public class CodeReading3 {
    public String toString() { return "North"; }
    public static void main (String [] args) {
        CodeReading3 s = new SubClass();
        System.out.println( s );
    }
}

class SubClass extends CodeReading3 {
    public String toString() { return "South"; }
}
```

4. What does it print after executing the following program? (4 points)

```
public class CodeReading4 {
    public static int[][] pickUpSomething (int[][] a, int x, int y) {
        int[][] result = new int[a.length][];
        for (int i = 0; i < a.length; i++) {
            int[] temp = new int[ a[i].length ];
            int count = 0;
            for (int n : a[i])
                if (x <= n && n <= y) temp[ count++ ] = n;
            int[] row = new int[count];
            System.arraycopy( temp, 0, row, 0, count );
            result[i] = row;
        }
        return result;
    }

    public static void main (String[] args) {
        int[][] b = { {1,-2,0,3}, {2,-4,0,1,8}, {0,0,10,0}, {0,18,0} };
        int[][] a = pickUpSomething( b, 1, 10 );
        for (int[] row : a)
            System.out.println( java.util.Arrays.toString( row ));
    }
}
```

Part VI: Programming (30 points)

1. (**Distance in 3-D space**) The formula for computing the distance between two points (x_1, y_1, z_1) and (x_2, y_2, z_2) in 3-dimension space is :

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}.$$

Write the static method distance to compute the distance between point p1 and p2. The square root of x can be computed using `Math.sqrt(x)`.

(TIPS: NO NEED TO WRITE ANY TESTING CODE with input/output!)

The program framework is given below:

```
public class Point {
    private double x, y, z;
    public Point(double x, double y, double z) {
        this.x = x; this.y = y; this.z = z;
    }
    public double getX() { return x; }
    public double getY() { return y; }
    public double getZ() { return z; }
```

```

    public static double distance(Point p1, Point p2) {
        ...
        /* YOUR CODE SEGMENT 1 HERE */
        ...
    };
}

```

2. (**Perfect number**) A perfect number is a positive integer that is equal to the sum of its proper divisors (factors). The smallest perfect number is 6, which is the sum of 1, 2 and 3. Other perfect numbers are 28, 496 and 8128.

Write a program called CheckPerfectNumber to check whether an input number from the console is a perfect number. For examples:

```

...>java CheckPerfectNumber
Enter the number to be checked: 15
15 is not a perfect number.

```

```

...>java CheckPerfectNumber
Enter the number to be checked: 28
Yes, proper factors are 1,2,4,7,14 and 1+2+4+7+14=28.

```

The program framework is given below:

```

import java.util.Scanner;
public class CheckPerfectNumber {

    public static void main(String[] args) {
        ...
        /* YOUR CODE SEGMENT 2.1 HERE */
        ...
        if (isPerfectNumber(n))
            System.out.printf( "Yes, %s.\n", perfectEvidence(n) );
        else
            System.out.printf( "%d is not a perfect number.", n);
    }

    public static boolean isPerfectNumber(int n) {
        ...
        /* YOUR CODE SEGMENT 2.2 HERE */
        ...
    }
}

```

```

private static String perfectEvidence(int n) {
    String factors = "1";
    String addends = "1";
    int factorLimit = (int)Math.sqrt(n+1);
    for (int i = 2; i <= factorLimit; i++) {
        if (n%i == 0) {
            factors += "," + i;
            addends += "+" + i;
        }
    }
    return String.format( "proper factors are %s and %s = %d",
        factors, addends, n
    );
}
}

```

3. (**Multiply two matrices**) Write a static method to multiply two matrices.

To multiply matrix **a** by matrix **b**, the number of columns in **a** must be the same as the number of rows in **b**, and the two matrices must have elements of the same or compatible types. Let **c** be the result of the multiplication. Assume the column size of matrix **a** is **n**. Each element c_{ij} is $a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + \dots + a_{in} \times b_{nj}$.

For example, a matrix **a[2][3]** multiplied by a matrix **b[3][4]** will get a matrix **c[2][4]** as the following formula:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \times \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \end{pmatrix}$$

where $c_{ij} = a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + a_{i3} \times b_{3j}$.

(TIPS: NO NEED TO WRITE ANY TESTING CODE with input/output!)

The method header is given below:

```

public static double[][] multiply(double[][] a, double[][] b) {
    ...
    /* YOUR CODE SEGMENT 3 HERE */
    ...
}

```