Week - 1 Long Descriptive Questions

1. What are the different types of comment symbols in Java?

• Single-line comments(//)

Single-line comments are denoted by // at the beginning of the comment.

Example - // Hello,World.

• Multi-line Comment(/*…..*/)

Multi-line comments are denoted by /* at the beginning and */at the end. They areused to write explanations of the code among many other uses.

Example - /* The above line of code instantiates the object obj */.

2. What are the data types supported in Java?

The primitive data types supported by Java are,

- Byte (8-bit Signed values).
- Short (16-bit Signed values).
- Integer (32-bit Signed values).
- Long (64-bit Signed values).
- Float (32-bit Floating point).
- Double (64-bit Floating point).
- Char (16-bit Unsigned values).

3. What is the difference between a Char in C/C++ and Char in Java?

Char in C&C++,

In C&C++, the character is denoted by the keyword char and it is stored in the size of 1 byteor 8-bits and it is stored in ASCII (American Standard Code for Information Interface) code.

Char in Java,

In Java, the character is denoted by the keyword char as same as C&C++ but the size is 2bytes or 16-bits and it is stored in Unicode instead of ASCII.

---

4. What are the different types of operators used in Java?

Java provides a rich operator environment. Most of the operators can be divided intofour categories, They are,

- Arithmetic Operators.
- Bitwise Operators.
- Relational Operators.
- Logical Operators.

Java also provides some special operators to handle some special situations. They are,

- Assignment Operators.
- Ternary Operator / The ? Operator.

Arithmetic Operators

The Arithmetic Operators are used in mathematical expressions in the same way that they areused in algebra.

Operator	Result
+	Addition (also unary plus).
-	Subtraction (also unary minus).
*	Multiplication.
/	Division.
%	Modulus.
++	Increment.
+=	Addition assignment.
-=	Subtraction assignment.

*=	Multiplication assignment.
/=	Division assignment.
%=	Modulus assignment.
- -	Decrement.

Bitwise Operators

Java defines several bitwise operators that can be applied to the integer types: long, int, short,char, and byte. These operators act upon the individual bits of their operands.

Operators	Result
~	Bitwise unary NOT.
&	Bitwise AND.
|	Bitwise OR.
^	Bitwise exclusive OR.
>>	Shift right.
>>>	Shift right zero fill.
<<	Shift left.
&=	Bitwise AND assignment.
|=	Bitwise OR assignment.
^=	Bitwise exclusive OR assignment.
>>=	Shift right assignment.
>>>=	Shift right zero fill assignment.
<<=	Shift left assignment.

Relational Operators

Relational operators determine the relationship that one operand has to the other. Specifically,they determine equality and ordering.

Operators	Result
= =	Equal to.
! =	Not Equal to.
>	Greater than.
<	Lesser than.
>=	Greater than or equal to.
<=	Lesser than or equal to

Logical Operators

The Boolean logical operators shown here operate only on boolean operands. Most of thebinary logical operators combine two boolean values to form a resultant boolean value.

Operator	Result
&	Logical AND.
|	Logical OR.
^	Logical XOR.
||	Short-circuit OR.
&&	Short-circuit AND.
!	Logical unary NOT
&=	AND assignment.
|=	OR assignment.
^=	XOR assignment.

Assignment Operator

The assignment operator is the single equal to symbol (=). The syntax of the assignmentoperator is,

Syntax,

variable = expression;

Exampe

a=b+c;

Ternary Operator / The ? Operator

Java includes a special ternary operator that can be replaced certain types of if-then-elsestatements. This operator is ?.

Syntax,

Example,

expression1?expression:expression3

```
int a = 5, b = 10;
int greaterNo = a>b ? a : b
```

5. Develop an Interest interface which contains simpleInterest and compInterest methods and a static final field of Rate 25%.

Aim:

To develop an interest interface which contain simple interest and compInterest methods and static final field of rate 25%.

Code:

```
import java.util.Scanner;

public class Main {
    static final double r = 25.0/100.0;
    public static void main(String[] args) {
        double principle, result1, result2, periodInYears;
        Scanner scan = new Scanner(System.in);
        System.out.println("Welcome, Please enter the principle amount:");
        principle = scan.nextDouble();
        System.out.println("Please enter the Period of interest in Years:");
        periodInYears = scan.nextDouble();
        result1 = simpleInterest(principle,periodInYears);
        System.out.println("Your Simple Interest is " +
                result1);
        result2 = compInterest(principle,periodInYears);
        System.out.println("Your Compound Interest is "+
                result2);
    }
    public static double simpleInterest(double prin, double


period){
        double res, interest;
        interest = prin*r;
        res = (interest*period)+ prin;
        return res;
```

```
    }
    public static double compInterest(double prin, double
            period){
        double res,interest;
        for (int i = 1; i<=period;i++){
            interest = prin*r;
            prin = prin+interest;
        }
        res = prin;
        return res;
    }
}
```

OUTPUT

```
Welcome, Please enter the principle amount: 1000
Please enter the Period of interest in Years: 5
Your Simple Interest is 2250.0
Your Compound Interest is 3051.7578125
```