Q-1: Explain the difference between if-else and switch in Java. When would you prefer one over the other?

Ans:

→ "if-else" is used when checking conditions with relational (>, <, ==) etc or logical (&&, ||) operators.

→ switch is used when a variable has multiple fixed values.

When to use?

→ Use if else for complex conditions

→ Use switch when checking a single variable with many possible values.

Q-2: Compare and contrast the "for," "while" and "do-while" loop in Java.

Ans:

for loop: When number of iteratons is known.

```
for (int i = 1; i<=5; i++)
  { system.out.println (i); }
```

While loop: When condition is checked before execution.

```
int i=1;
while (i<=5)
  { System.out.println(i);
    i++;
  }
```

do-while: When condition is checked after execution (executes at least once)

```
int i=1
  do { system.out.println(i);
       i++;}
     while (i<=5);
```

Q-3: Explain the purpose of each part of a for loop's syntax: initialization, condition and iteration expression.

Ans → **initialization:** int i=1; → Runs once, set the start value.

→ **condition:** i<=5; → Checked before each iteration

→ **Iteration Expression:** i++ → Updates loop variable after each iteration.

Q-4: Define a method in Java and explain its components.

Ans:

→ A method is a block of code that performs a task.

→ Components:

Return type: void, int

Method name: int sum()

Parameters: (Optional)

Body: (Code inside { })

Ex:
```
Public void greet ()
        { system.out. println ("Hello");
```

Q-5: Define "method overloading" concept in Java with an example.

Ans: Method overloading means multiple methods with the same name but different parameters.

→ Java decides which method to call based on the number ofr type of arguments

```
class OverloadExample {
 void display (int a)
  { System.out.println ("Integer : " + a); }
void display (string s)
   { system.out.println ("String : " + s); }
```

(Ans-6):

Benefits of overloading:

→ Make the code more readable

→ Increase reusability

→ Improves maintainability by reducing redundant code.

Drawbacks:

→ Can be confusing if not used properly.

→ May increase complexity in large projects

→ If overloading with many different data types Java must determine the correct method at runtime