Java Programming Assignment

Section 1: Java Data Types

1. **What are the different primitive data types available in Java?**
   There are 8 different types of primitive data:
   byte, short, int, long, float, double, char, Boolean.

2. **Explain the difference between primitive and non-primitive data types in Java.**
   Primitive data types store simple values and are predefined by Java.
   Non-primitive data types are objects, store references, and are created from classes.

3. **Write a Java program that demonstrates the use of all primitive data types.**

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

byte b = 15;
short s = 200;
int i = 1500;
long l = 10000L;
float f = 20.5f;
double d = 30.99;
char c = 'A';
boolean bool = true;

System.out.println(b);
System.out.println(s);
System.out.println(i);
System.out.println(l);
System.out.println(f);
System.out.println(d);
System.out.println(c);
System.out.println(bool);
} }
```

4. **What is type casting? Provide an example of implicit and explicit casting in Java.**
   Type Casting is conversion of one data type into another data type.

   ->***Implicit casting*** (widening): smaller data type to bigger data type, it is done automatically.
   Example: int i = 100;
   long l = i;

   ->***Explicit casting*** (narrowing): bigger data type to a smaller data type, IT is done manually.
   Example: double d = 200.5;
   int i = (int) d;
5. **What is the default value of each primitive data type in Java?**
   byte = 0, short = 0, int = 0, long = 0L, float = 0.0f, double = 0.0d, char = '\u0000', boolean = false.

Section 2: Java Control Statements

1. **What are control statements in Java? List the types with examples.**
   Control statements control the flow of execution in a program.
   Types of Control Statements are :
   - Decision making: if, if-else, switch
   - Looping: for, while, do-while
   - Jumping: break, continue, return

   Examples:
   - if (condition) {
     // code
     }
   - for (int i=0; i<10; i++) {
     //code
     }
   - break; inside loop to exit loop.
2. **Write a Java program to demonstrate the use of if-else and switch-case statements.**

```
public class ControlStatementsDemo {
  public static void main(String[] args) {
    int day = 5;
    if (day == 6 || day == 7) {
      System.out.println("It's a weekend");
    } else if (day >= 1 && day <= 5) {
      System.out.println("It's a weekday");
    } else {
      System.out.println("Invalid day number");
```

```
        }
        //  ************************
        // switch CASE to print Day
        switch (day) {
          case 1:
            System.out.println("Monday");
            break;
          case 2:
            System.out.println("Tuesday");
            break;
          case 3:
            System.out.println("Wednesday");
            break;
          case 4:
            System.out.println("Thursday");
            break;
          case 5:
            System.out.println("Friday");
            break;
          case 6:
            System.out.println("Saturday");
            break;
          case 7:
            System.out.println("Sunday");
            break;
          default:
            System.out.println("Invalid day number");
        }
      }
    }
```

3. **What is the difference between break and continue statements?**
   break: It Exits the entire loop or switch statement immediately.
   continue: It Skips the current iteration and moves to the next iteration of the loop.
4. **Write a Java program to print even numbers between 1 to 50 using a for loop.**

```
public class EvenPrintDemo {
public static void main(String[] args) {
for (int i = 1; i <= 50; i++) {
if (i % 2 == 0) {
System.out.print(i + " ");
}
}
}
```

}

5. **Explain the differences between while and do-while loops with examples.**
   while loop checks condition first, executes loop only if condition is true.
   do-while loop executes loop body first, then checks condition (executes at least once).

Section 3: Java Keywords and Operators

1. **What are keywords in Java? List 10 commonly used keywords.**
   Keywords are Reserved words in Java with predefined meaning; they cannot be used as identifiers.
   Common keywords:
   public, static, class, final , void, if, else, for, while, return

2. **Explain the purpose of the following keywords: static, final, this, super.**
   static: belongs to the class, shared by all instances.
   final: makes variable constant or prevents method/class modification or overriding.
   this: refers to the current object instance.
   super: refers to parent class object.
3. **What are the types of operators in Java?**
   - Arithmetic (+, -, *, /, %)
   - Relational (==, !=, >, <, >=, <=)
   - Logical (&&, ||, !)
   - Assignment (=, +=, -=)
   - Unary (++, --, !)
   - Bitwise (&, |, ^, ~, <<, >>)
4. **Write a Java program demonstrating the use of arithmetic, relational, and logical operators.**

```
public class OperatorsDemo {
public static void main(String[] args) {
int a = 10, b = 20;
// Arithmetic operator
System.out.println("a + b = " + (a + b));
System.out.println("b - a = " + (b - a));
System.out.println("a * b = " + (a * b));
// Relational operator
```

```
System.out.println("a == b: " + (a == b));
System.out.println("a < b: " + (a < b));
// Logical operator
System.out.println("(a < b) && (a > 5): " + ((a < b) && (a > 5)));
System.out.println("(a > b) || (b > 15): " + ((a > b) || (b > 15)));
}
}
```

5. **What is operator precedence? How does it affect the outcome of expressions?**
   Operator precedence determines the sequence in which different operators in an expression are processed. Operators with higher precedence are evaluated first, which influences the outcome of the expression. Using parentheses allows you to explicitly specify the order of evaluation, overriding default precedence rules.

Additional Questions

Java Data Types

6. **What is the size and range of each primitive data type in Java?**
   byte: 1 byte, range: -128 to 127
   short: 2 bytes, range: -32,768 to 32,767
   int: 4 bytes, range : -2^{31} to $2^{31} - 1$
   long: 8 bytes, range: -2^{63} to $2^{63} - 1$
   float: 4 bytes, range : approx ±3.4e−38 to ±3.4e+38
   double: 8 bytes, range: approx ±1.7e−308 to ±1.7e+308
   char: 2 bytes, range: 0 to 65,535 (Unicode characters)
   boolean: 1 bit, values true or false

7. **How does Java handle overflow and underflow with numeric types?**
   Java does not raise errors on numeric overflow or underflow; instead, values wrap around according to two's complement arithmetic.
8. **Write a program to convert a double value to an int without data loss.**

```
public class ConvertDoubleToInt {
public static void main ( String[] args) {
double d = 100.0;
int i = (int) d; // works without data loss only if decimal part is zero
System.out.println(i);
}
}
```

9. **What is the difference between char and String in Java?**
   - Char holds a single Unicode character.
   - String is a sequence of characters (object).
10. **Explain wrapper classes and their use in Java.**

Wrapper classes (e.g., Integer, Double) wrap primitive types into objects; used in collections and
for converting between primitives and objects.

Java Control Statements

6. Write a Java program using nested if statements.

```
int num = 20;
if (num > 0) {
if (num % 2 == 0) {
System.out.println("Positive even number");
}
}
```

7. Write a Java program to display the multiplication table of a number using a loop.

```
int num = 5;
for (int i = 1; i <= 10; i++) {
System.out.println(num + " x " + i + " = " + (num * i));
}
```

8. How do you exit from nested loops in Java?
   Using break statement
9. Compare and contrast for, while, and do-while loops.
   - for: best when number of iterations known.
   - while: condition checked before loop body.
   - do-while: executes loop body at least once, condition checked after.

10. Write a program that uses a switch-case to simulate a basic calculator.

```
import java.util.Scanner;

public class Calculator {
  public static void main(String[] args) {
    try (Scanner sc = new Scanner(System.in)) {
      System.out.print("Enter a: ");
      int a = sc.nextInt();

      System.out.print("Enter b: ");
      int b = sc.nextInt();

      System.out.print("Choose (+, -, *, /): ");
      char op = sc.next().charAt(0);

      switch (op) {
```

```
            case '+':
                System.out.println("Sum: " + (a + b));
                break;
            case '-':
                System.out.println("Difference: " + (a - b));
                break;
            case '*':
                System.out.println("Product: " + (a * b));
                break;
            case '/':
                if (b != 0) {
                    System.out.println("Quotient: " + (a / b));
                } else {
                    System.out.println("Error: Division by zero is not allowed.");
                }
                break;
            default:
                System.out.println("Invalid operator");
        }
    }
  }
}
```

Java Keywords and Operators

6. **What is the use of the `instanceof` keyword in Java?**
   Checks if an object is an instance of a specific class or interface.
7. **Explain the difference between `==` and `.equals()` in Java.**
   == : compares references (memory address).
   .equals() : compares content (logical equality).
8. **Write a program using the ternary operator.**
   public class TernaryOperatorDemo{
   public static void main(String[] args) {
   int num = 10;
   String result = (num % 2 == 0) ? "Even" : "Odd";
   System.out.println(result);
   }}
9. **What is the use of `this` and `super` in method overriding?**
   this refers to current class method/variable.
   super calls overridden parent class method.

10. Explain bitwise operators with examples.

```
int a = 5; // 0101
int b = 3; // 0011
System.out.println(a & b); // 1 (0001)
System.out.println(a | b); // 7 (0111)
System.out.println(a ^ b); // 6 (0110)
System.out.println(~a); // -6 (two's complement)
System.out.println(a << 1); // 10 (1010)
System.out.println(a >> 1); // 2 (0010)
```