Java Programming Assignment

Section 1: Java Data Types

1. What are the different primitive data types available in Java?
    - byte – 8-bit integer
    - short – 16-bit integer
    - int – 32-bit integer
    - long – 64-bit integer
    - float – 32-bit floating-point
    - double – 64-bit floating-point
    - char – 16-bit Unicode character
    - boolean – true or false

2. Explain the difference between primitive and non-primitive data types in Java.

Primitive	**Non-Primitive**
Directly in memory	References to objects
int, double, char	String, Arrays, Objects
Faster	Slower

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

```
public class PrimitiveTypes {
    public static void main(String[] args) {
        byte b = 100;
        short s = 20000;
        int i = 123456;
        long l = 123456789L;
        float f = 3.14f;
        double d = 123.456;
        char c = 'A';
        boolean bool = true;

        System.out.println("byte: " + b);
        System.out.println("short: " + s);
        System.out.println("int: " + i);
        System.out.println("long: " + l);
        System.out.println("float: " + f);
        System.out.println("double: " + d);
```

```
        System.out.println("char: " + c);
        System.out.println("boolean: " + bool);
    }
}
```

4. What is type casting? Provide an example of implicit and explicit casting in Java.
   Type casting convert a variable of one type to another.

   - **Implicite casting**
     int a = 10;
     double b = a; // int to double

   - **Explicit casting**
     double x = 9.78;
     int y = (int) x; // double to int

5. What is the default value of each primitive data type in Java?

Data type	Default Value
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
boolean	false

Section 2: Java Control Statements

1. What are control statements in Java? List the types with examples.
   - **Conditional**: if, if-else, switch
   - **Looping**: for, while, do-while
   - **Branching**: break, continue, return
   -

2. Write a Java program to demonstrate the use of if-else and switch-case statements.

```
public class ControlStatements{
    public static void main(String[] args) {
        int number = 5;

        if (number > 0) {
            System.out.println("Positive number");
        } else {
            System.out.println("Non-positive number");
        }

        int day = 3;
        switch (day) {
            case 1: System.out.println("Monday");
                    break;
            case 2: System.out.println("Tuesday");
                    break;
            case 3: System.out.println("Wednesday");
                    break;
            default: System.out.println("Invalid day");
        }
    }
}
```

3. What is the difference between break and continue statements?
    - **break**: exits the loop or switch.
    - **continue**: skips the current iteration and moves to the next loop cycle.
4. Write a Java program to print even numbers between 1 to 50 using a for loop.

```
public class EvenNumbers {
    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	do-while
Condition checked before	Runs once before checking
When loop may not run at all	Ensure it runs at least once

Section 3: Java Keywords and Operators

1. What are keywords in Java? List 10 commonly used keywords.

    - Class
    - Public
    - Static
    - Void
    - Int
    - If
    - Else
    - For
    - Return
    - New

2. Explain the purpose of the following keywords: static, final, this, super.
    - **static**: Belongs to the class, not instance.
    - **final**: Constant value or prevents method/ class override.
    - **this**: Refers to current object.
    - **super**: Refers to parent class.
3. What are the types of operators in Java?
    - **Arithmetic**: +, -, *, /, %
    - **Relational**: ==, !=, >, <, >=, <=
    - **Logical**: &&, ||, !
    - **Bitwise**, Assignment, Unary, Ternary, etc.
4. Write a Java program demonstrating the use of arithmetic, relational, and logical operators.

```
public class Operators {
    public static void main(String[] args) {
        int a = 10, b = 5;
        System.out.println("Arithmetic: " + (a + b));
        System.out.println("Relational: " + (a > b));
        System.out.println("Logical: " + ((a > b) && (b > 0)));
    }
}
```

5. What is operator precedence? How does it affect the outcome of expressions?
   Operator precedence determines the order of operations.
   int result = 10 + 5 * 2;
   // result = 20, * has higher precedence than +

Additional Questions

Java Data Types

6. What is the size and range of each primitive data type in Java?

Type	Size	Range
byte	8-bit	-128 to 127
short	16-bit	-32,768 to 32,767
Int	32-bit	-2^31 to 2^31-1
Long	64-bit	-2^63 to 2^63-1
boolean	1-bit	true/false

7. How does Java handle overflow and underflow with numeric types?
    - byte b = 127;
      b++; // Overflow: becomes -128

8. Write a program to convert a double value to an int without data loss.

    - double d = 12.56;
      int i = (int) Math.round(d);
      System.out.println(i); // Output: 13

9. What is the difference between char and String in Java?
    - char: Single 16-bit character.
    - String: Sequence of characters (object).
    -

10. Explain wrapper classes and their use in Java.

- Each primitive has a corresponding wrapper:
- int → Integer, double → Double, etc.
- Used in collections, null support, type conversion

Java Control Statements

6. Write a Java program using nested if statements.

```
int num = 15;
if (num > 0) {
    if (num % 2 == 0)
        System.out.println("Positive even");
    else
        System.out.println("Positive odd");
}
```

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

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

8. How do you exit from nested loops in Java?

```
outer:
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= 5; j++) {
        if (i * j > 10)
            break outer;
        System.out.println(i + " " + j);
    }
}
```

9. Compare and contrast for, while, and do-while loops.

For loop	**While loop**	**Do-while loop**
At the top of the loop	Before the loop	Before the loop
At the top (before execution)	At the top (before execution)	At the bottom(before execution)
0 times	0 times	At least once
Known iteration count	Unknown count, condition-based	When loop must run at least once

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

```java
import java.util.Scanner;

public class Calculator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first number: ");
        double a = sc.nextDouble();
        System.out.print("Enter operator (+, -, *, /): ");
        char op = sc.next().charAt(0);
        System.out.print("Enter second number: ");
        double b = sc.nextDouble();

        switch (op) {
            case '+': System.out.println("Result: " + (a + b)); break;
            case '-': System.out.println("Result: " + (a - b)); break;
            case '*': System.out.println("Result: " + (a * b)); break;
            case '/':
                if (b != 0)
                    System.out.println("Result: " + (a / b));
                else
                    System.out.println("Error: Division by zero");
                break;
            default: System.out.println("Invalid operator");
        }
    }
}
```

Java Keywords and Operators

6. What is the use of the `instanceof` keyword in Java?
   instanceof is used to test whether an object is an instance of a specific class or subclass.
   - String s = "Hello";
     System.out.println(s instanceof String); // true

7. Explain the difference between `==` and `.equals()` in Java.
   - **==:** compares reference (memory address) of two objects.
   - .**equals**(): compares the contents (values) of two objects.
   -

     String s1 = new String("hello");
     String s2 = new String("hello");

     System.out.println(s1 == s2);   // false (different objects)
     System.out.println(s1.equals(s2)); // true (same content)

8. Write a program using the ternary operator.

```
public class TernaryDemo {
    public static void main(String[] args) {
        int a = 10, b = 20;
        String result = (a > b) ? "a is greater" : "b is greater";
        System.out.println(result);
    }
}
```

9. What is the use of `this` and `super` in method overriding?

```
class Animal {
    void sound() {
        System.out.println("Animal makes sound");
    }
}

class Dog extends Animal {
    void sound() {
        super.sound();  // Call parent method
        System.out.println("Dog barks");
    }
}

public class Test {
    public static void main(String[] args) {
        Dog d = new Dog();
        d.sound();
    }
}
```

Use of this:

```
class Example {
  int x;
  Example(int x) {
    this.x = x;  // distinguish between parameter and instance variable
  }
}
```

10. Explain bitwise operators with examples.

operator	meaning
&	AND
^	XOR
~	NOT
<<	Left shift
>>	Right shift

Code:

```
public class BitwiseDemo {
    public static void main(String[] args) {
        int a = 5;  // 0101
        int b = 3;  // 0011

        System.out.println("a & b = " + (a & b));  // 0001 = 1
        System.out.println("a | b = " + (a | b));  // 0111 = 7
        System.out.println("a ^ b = " + (a ^ b));  // 0110 = 6
        System.out.println("~a = " + (~a));        // inverts all bits
        System.out.println("a << 1 = " + (a << 1)); // 1010 = 10
        System.out.println("a >> 1 = " + (a >> 1)); // 0010 = 2
    }
}
```