java

Section 1:

1. Primitive data types are basic data types that are built into the programming language, while reference data types are data types that are defined by the programmer. Primitive data types are passed by value, while reference data types are passed by reference. Examples of primitive data types include int, float, and boolean, while examples of reference data types include arrays, classes, and interfaces.
2. The scope of a variable refers to the part of the program where the variable can be accessed. In Java, there are two types of variables: local variables and global variables. Local variables are declared inside a method or block and can only be accessed within that method or block. Global variables, also known as instance variables, are declared outside of any method or block and can be accessed by any method or block within the same class.
3. Initialization of variables is required to assign an initial value to the variable. If a variable is not initialized, it will contain a default value, which may not be the desired value. Initializing variables also helps to avoid errors and bugs in the program.
4. Static variables are variables that are shared by all instances of a class and are initialized only once, while instance variables are variables that are unique to each instance of a class. Local variables are variables that are declared inside a method or block and have a limited scope.
5. Widening casting is the conversion of a smaller data type to a larger data type, while narrowing casting is the conversion of a larger data type to a smaller data type. Widening casting is done automatically by the compiler, while narrowing casting requires an explicit cast operator.
6.

TYPE	SIZE (IN BYTES)	DEFAULT	RANGE
boolean	1 bit	false	true, false
char	2	'\u0000'	'\u0000' to '\uffff'
byte	1	0	-128 to 127

short	2	0	-32,768 to 32,767
int	4	0	-2,147,483,648 to 2,147,483,647
long	8	0L	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4	0.0f	1.4E-45 to 3.4028235E38
double	8	0.0d	4.9E-324 to 1.7976931348623157E308

1. In Java programming, a package is a collection of related classes and interfaces that are grouped together. Packages help to organize code and prevent naming conflicts. Packages also provide access control by allowing classes to be declared as public, protected, or private.
2. Using Java packages is important because it helps to organize code, prevent naming conflicts, and provide access control. Packages also make it easier to reuse code and share code between projects

section 2.

1.

```
import java.util.Scanner;

public class SurnameAndAge {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter your surname: ");
        String surname = input.nextLine();

        System.out.print("Enter your age: ");
        int age = input.nextInt();

        System.out.println("The number of characters in your surname is " + surname.length());

        if (age % 2 == 0) {
            System.out.println("Your current age is an even number");
        } else {
            System.out.println("Your current age is an odd number");
        }
    }
}
```

2.

```
import java.util.Scanner;

public class AverageMarks {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter marks for unit 1: ");
        double unit1 = input.nextDouble();

        System.out.print("Enter marks for unit 2: ");
        double unit2 = input.nextDouble();

        System.out.print("Enter marks for unit 3: ");
        double unit3 = input.nextDouble();

        System.out.print("Enter marks for unit 4: ");
        double unit4 = input.nextDouble();

        System.out.print("Enter marks for unit 5: ");
        double unit5 = input.nextDouble();

        double average = (unit1 + unit2 + unit3 + unit4 + unit5) / 5;

        System.out.printf("The average marks is %.2f", average);
    }
}
```

3.

```
```