22/03170

SEC 1

1.-Primitive data types are basic data types that are built in to a programming languages integers, While reference data types are defined by a programmer.eg strings.

-Primitive data types are stored directly in memory while, reference data types are stored indirectly in memory.

-Primitive data types has a default value assigned by the programming language eg an integer has a default value of 0 while, reference data type has a default value of null, the reference does not currently point to any valid object or data.

2. The scope of a variable refers to the region where the variable is recognized and can be used.

Local variable	**Global variable**
Limited visibility and are accessible only within the block of code where they are declared.	Are visible to all functions and blocks within the program, therefore accessible throughout the execution of the program.
Are temporary and exist only during the execution of the block of code.	They have global scope and can be accessed and modified from any part of the program, including functions, loops, and other blocks of code.
Are declared within a specific block of code, eh function or loop.	Are declared outside of any specific block of code, typically at the top of a program.
Once the block of code is exited, the variable is destroyed and its memory released.	Modification of the variable in one part of the program affects its value in other parts of the program.

3. Initialization of variables is required since:

-it helps remove bugs and errors caused by using uninitialized variables.

-It prevents a variable to contain a random/ undefined value.

-Allows you to provide default values for variables.

-By initializing a variable, a consistent starting point is established.

4. Static variables also known as class variables are associated with a class rather than with instances or objects of the class. While instance variables also known as member variables/ object variables are unique to each instance or object of class. Local variables are declared with a method, constructor or block of code that have limited scope.

5. Widening casting (automatically) is converting a smaller data type to a larger data type. While

Narrowing Casting (Manually) is converting a larger data type to a smaller size data type.

6.

TYPE	SIZE IN BYTES	DEFAULT	RANGE
Boolean	1 bit	false	True, false
Char	2	‘0\	\0000’\ to \’ffff’’
Byte	1	0	-128 to +127
Short	2	0	-215 to +215-1
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	00.0f	-3.4+38 to +3.4+3.8
Double	8	0.0d	-1.8E+308 to +1.8E+308

7. A package presents a directory that contain relate group of classes and interface.

8. – Packages are useful to arrange related classes and interface into a group. This makes all the classes and interface performing the same task to put together in the same package.

- Packages hide the classes and interfaces in a separate subdirectory, so that accidental deletion of classes and interfaces will not take place.

- The classes and interfaces of a package are isolated from the classes and interfaces of another package. This means that we can use same names for classes of two different classes.

- A group of packages is called a library. The classes and interface of a package are like books in a library and can be reused several times. This reusability nature of packages makes programming easy.

SEC 2

Please copy and paste to your browser so as to view the codes.

1. /*The program uses the Scanner class from the java.util package to read user input.

The user is prompted to enter their surname .The user is prompted to enter their age

The length() method is used to determine the number of characters in the surname. The number of characters in the surname is checked. The age is either even or odd.

*/

```
import java.util.Scanner;

public class SurnameAndAge {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Asking for surname

System.out.print("Enter your surname: ");

String surname = scanner.nextLine();
```

```
// Asking for age

System.out.print("Enter your current age: ");

Java Assignment 1 4

int age = scanner.nextInt();

// Printing the number of characters in the surname

int surnameLength = surname.length();

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

// Checking if the age is even or odd

String evenOdd = (age % 2 == 0) ? "even" : "odd";

System.out.println("Your current age is an " + evenOdd + " number");

scanner.close();

}

}
```

2. /*The program uses the Scanner class from the java.util package to read user input. The user is prompted to enter the marks for each of the five units using a for loop. The double values entered by the user are read. The sum of the marks is calculated by adding each input value to the sum variable. the average is calculated by dividing the sum by 5 (the number of units) and displayed with 2 decimal places.

```
*/

import java.util.Scanner;

public class MarksAverage {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Asking for marks of five units

System.out.println("Enter the marks of five units:");

double sum = 0;

for (int i = 1; i <= 5; i++) {

System.out.print("Unit " + i + ": ");

double marks = scanner.nextDouble();
```

```
sum += marks;
}
// Calculate average
double average = sum / 5;
// Display average with two decimal places
System.out.printf("The average marks is: %.2f%n", average);
scanner.close();
}
}
```

3. /*The program uses the Scanner class from the java.util package to read user input.
The user is prompted to enter an integer
A for loop is used to iterate through integers in the range of 0-9.
The % operator is used to check whether the number is divisible by the current integer in the loop.
If the number is divisible by an integer, a message is printed
*/

```
import java.util.Scanner;
public class DivisibilityTest {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
```

Java Assignment 1 5

```
// Ask for the integer to be tested
System.out.print("Enter an integer: ");
int number = scanner.nextInt();
// Checking divisibility by integers in the range of 0-9
for (int i = 0; i <= 9; i++) {
if (number % i == 0) {
System.out.println("The number is divisible by " + i);
}
```

```
}
scanner.close();
}
}
```

4. /*The program defines the starting range as 71 and the ending range as 150.

The % operator is used to check whether the current number is divisible by 2, 3, or 7,

If the number is a multiple of 2, 3, or 7, it is printed using System.out.println.

The loop continues until the end of the range is reached. The program displays all the multiples of 2, 3, and 7 within the range 71 to 150

*/

```
public class MultiplesInRange {
public static void main(String[] args) {
int startRange = 71;
int endRange = 150;
System.out.println("Multiples of 2, 3, and 7 within the range 71 to 150:");
for (int i = startRange; i <= endRange; i++) {
if (i % 2 == 0 || i % 3 == 0 || i % 7 == 0) {
System.out.println(i);
}
}
}
}
```

5. /*The program uses the Scanner class from the java.util package to read user input.

The user is prompted to enter the first number, the operation (+, -, *, or /), and the second number.

A switch statement is used to perform the selected operation based on the user's input.

The result of the operation is stored in the result variable.

If the user attempts to divide by zero, an error message is displayed, and the program exits.

```
If an invalid operation is entered, an error message is displayed, and the program exits.
The result is displayed
*/
import java.util.Scanner;
public class Calculator {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
// Ask for the first number
System.out.print("Enter the first number: ");
double num1 = scanner.nextDouble();
// Ask for the operation
System.out.print("Enter the operation (+, -, *, /): ");
String operation = scanner.next();
// Ask for the second number
System.out.print("Enter the second number: ");
double num2 = scanner.nextDouble();
double result = 0;
// Perform the operation and calculating the result
switch (operation) {
case "+":
result = num1 + num2;
break;
case "-":
result = num1 - num2;
break;
case "*":
result = num1 * num2;
break;
case "/":
```

```
if (num2 != 0) {
result = num1 / num2;
} else {
System.out.println("Error: Division by zero is not allowed.");
System.exit(0);
}
break;
default:
System.out.println("Error: Invalid operation entered.");
System.exit(0);
}
// Display the result
System.out.println("Result: " + result);
scanner.close();
}
}
```