

Assignment 1

Date _____
Page _____

1 Which of the following arithmetic expressions are valid?



valid : 23.45, 12D.13

invalid : 23.45d

invalid : 12.3e-1.2

2 Write Java assignment statements to evaluate the following as equations :

→ float area = (float) Math.PI * (r + r) * (2 * (float) Math.PI * r * b);

float torque = (2 * m1 + m2 * 2) / (m1 + m2) * g

double temp = a * a + b * b - 2 * a * b * Math.cos(x);

double side = Math.sqrt(temp);

float energy = mass * (acceleration * height + velocity * velocity / 2);

3 Find error, if any and rectify them :

→ a. float x = 0.5f, y = 2.0f, z = 5.75f;

b. int m = ++a * 5;

c. double q = Math.sqrt(100);

d. float p = x/y;

e. int s = 5;

f. ans = (b+t) - c * 2;

What is the output of print statement below?

What will be the output of the following program?

Q4) Determine the value of each of the following logical & expressions if $a=5$, $b=10$ & $c=-6$

→ a. $a>b \ \&\& \ a<c = \text{false}$

b. $a < b \ \&\& \ a > c = \text{true}$

c. $a == c \ \text{||} \ b > a = \text{true}$

d. $b > 15 \ \&\& \ c < 0 \ \text{||} \ a > 0 = \text{true}$

In the following code the expected value is 78 but it returns the value 89. Modify the code :

```
public static void calculate()
```

{

```
    int ans = 42 + 45 - 48 - 5 - 15 + 20 * 2;  
    System.out.println("Value of ans: " + ans);
```

y

→ public static void calculate()

{

```
    int ans = (42 + (45 - 48 - 5 - 15) + 20) * 2;
```

```
    System.out.println("Value of ans: " + ans);
```

y

6) Provide the reason for declaring main method as a static in Java.

→ The main() method in C++, C# & Java are static because they are linked by the runtime engine without having to instantiate any objects then the code in the body of

`main()` will do the rest.

Java program's main method has to be declared static because keyword static allows main to be called without creating an object of the class in which the main method is defined.

- 7 Determine whether the following are true or false:
- a. When if statements are nested, the last else gets associated with the nearest if without an else = true
 - b. One if can have more than one else clause = false
 - c. A switch statement can always be replaced by the series of if--else statements = true
 - d. A switch expression can be of any type = false
 - e. A program stops its execution when a break statement is encountered = false

8

Difference between next() and nextLine()

→

next()

nextLine()

- A method of Scanner class in java.util package that returns a string object which is a complete token of Scanner object.

A method of Scanner class in java.util package that returns a string which corresponds to the skipped line of the Scanner object.

- Capable of reading user inputs until receiving a space.

Capable of reading user inputs until pressing the enter key or receiving a new line.

- It is not possible to read words with spaces using next().

It is possible to read the words with spaces using the nextLine().

9

Write a detailed note on Scanner class.

Definition, Usage, Work, Advantage, Problem with Scanner class.

- Scanner class in Java is found in the java.util package. Java provides various ways to read input from the keyboard, the java.util.Scanner class is one of them.

- The java Scanner class breaker input into tokens using a delimiter which is whitespace by default. It provides many methods to read

and parse various primitive values.

- The Java Scanner class is widely used to parse text for strings and primitive types using a regular expression. It is the simplest way to get input in java. By the help of Scanner Scanner in java, we can get input from the user in primitive types such as int, long, double, byte, float, short etc.

Java Scanner class Declaration

```
public final class Scanner extends Object  
implements Iterator<String>
```

To get the instance of java Scanner which reads input from the user we need to pass the input stream (System.in) in the constructor of Scanner class.

Eg -

```
Scanner in = new Scanner (System.in);
```

In what ways does a switch statement differ from an if statement?

- A switch statement is usually more efficient than a set of nested ifs. Deciding whether to use if-then-else statements or a switch statement is based on reliability and the expression that the statement is testing.

1. Checking the testing expression:

An if - then - else statement can test expressions based on ranges of values or conditions, whereas a switch statements test expressions based only on a single integer, enumerated value or string object.

2. Switch better for multi-way branching:

When compiler a switch statement, it will inspect each of the case constants and create a "jump table" that is used for selecting the path of execution depending on the value of the expression. The compiler can do this because it knows that the case constants are all the same type and simply must be compared for equality. With the switch statement expression, while in case of if expression, the compiler ~~uses~~ no such knowledge.

3. Speed:

A switch statement might prove to be faster than if its provided number of cases are good. If there are only four cases, it might not effect the speed in any case. If a switch contains more than five items, its implemented using a lookup table or a hash list.

4. Clarity in readability

A switch looks much cleaner when you have to compile cases. It's ~~more~~ quite ^{difficult} to write too.

(*) advantages

11 Compare in terms of functions

a. while and do...while

while : The iterations do not occur if, the condition at the first iteration, appears false.

do...while : The iteration occurs at least once even if the condition is false at the first iteration.

b. while and for

while : The while loop will continue to run infinite number of time until the condition is met.

for : This is preferable when we know exactly how many times the loop will be repeated.

c. break and continue

break : When a break statement is encountered, it terminates the block and gets the control out of the switch or loop.

continue : When a continue statement is encountered, it gets the control to the next iteration of the loop.

12 What is an empty statement? Explain its usefulness.

→ As the name suggested, this statement does not contain anything other than a semicolon (;

You can use it in loops, and conditions where you aren't supposed to do anything.

empty statement in if-else :

```
java -jar final if (i==0)
```

```
    {  
        System.out.println ("i");  
    }  
    else  
    {  
        System.out.println ("j");  
    }  
}
```

14 Define array. How to initialize an array of float with 20+45 a size? Write a snippet codes to search specific element from an array.

→ Array is a fundamental construct in java that allows you to store and access large number of values conveniently with similar type.

```
float num [] = new float [20];
```

```
import java.util.*;  
public class FindNumberInarray{  
}
```

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

```

int m, loop;
Scanner sc = new Scanner(System.in);
System.out.println("Enter total number of element");
m = sc.nextInt();
System.out.println("Enter array elements");
int arr[] = new int[m];
for (loop = 0; loop < m; loop++) {
    System.out.print("Enter element " + (loop + 1) + ":");
    arr[loop] = sc.nextInt();
}
int num;
System.out.println("Enter element to search:");
num = sc.nextInt();
int index = -1;
for (loop = 0; loop < m; loop++) {
    if (arr[loop] == num) {
        index = loop;
        break;
    }
}
if (index == -1)
    System.out.println("Element not found");

```

```
else
    System.out.println("Element found at index " + index);
}
}
```

15 State features of an array. Consider a scenario where a person wants to create one integer type of array which has three rows, and having three, four & two column respectively. Draw a structure of array and write a program to create specified array and take element input from user and display all elements of an array.

→ Features of array

1. Fixed Length
2. Fast access
3. An array can hold primitives
4. An array of primitives stores value of the primitives
5. An array of objects stores only the reference to the object.
6. An array itself is actually an object.

Array structure

	0	1	2	3	← 4 columns
↑	0				← 1 row having 3 columns
↑	1				← 2 rows having 3 columns
↓	2				← 3 rows having 2 columns
↑					
	3 rows ← columns →				

```

class Array {
    public static void main (String [] args) {
        int [][] arr = new int [3] [];
        arr[0] = new int [] {1,2,3};
        arr[1] = new int [] {4,5,6,7};
        arr[2] = new int [] {8,9};
        for (int [] row : arr)
            System.out.println(Arrays.toString(row));
    }
}

```

- 16 what are objects? How are they created from a class?
- An entity that has state and behaviour is known as an object. E.g. chair, bike, pen, laptop, table etc. It can also be physical or logical.
- A class is a template from which individual objects are created. Following is a sample of a class. A class provides the template for objects; you create an object from a class.
 - Each of the following statements taken from the CreateObjectDemo assigns it to a variable.

Point originOne = new Point(23, 94);
Rectangle rectOne = new Rectangle(originOne, 100, 200);

- The first line creates an object of the Point class, and the second line creates an object of the Rectangle class.

17 Compare and contrast overloading and overriding methods.

→ Overloading Overriding

Method overloading is used to increase the readability of the program.
Method overriding is used to provide the specific implementation of the method that is already provided by its super class.

- Method overloading is performed within a class in two classes that have an IS-A relationship.

In case of method overloading, parameter must be different.

Ex. Compile-time polymorphism Ex. Run-time polymorphism.

- In java, method overriding can't be performed by changing return type or covariant in return type of the method only. Return type can be same or different in method overloading. But you must have to change the parameter.

18. What are the applications of wrapper classes?

- 1. They convert primitive data types into objects. Objects are needed if we wish to modify the arguments passed into a method.
- 2. The classes in `java.util` package handles only objects and hence wrapper classes help in this case also.
- 3. Data structures in the collection framework, such as `ArrayList` and `Vector`, store only objects and not primitive types.
- 4. An object is needed to support synchronization in multithreading.

19. Write a detailed note on this keyword.

```
→ class Sample {  
    int mark; rank;  
  
    sample (int mark, rank) {  
        mark = mark;  
        rank = rank;  
    }  
}
```

- In the constructor method above, how does body statements distinguish between method parameter and fields having the same name?
- This is where the `this` reference step in. Every method of a class has access to the keyword `this`.
- The reference `this` holds the address of the current object on which the method was invoked.
- The `this` reference will be passed to every methods as a hidden argument.
- Thus, the constructor has an implicit `this` argument.
- In the case of name clashing between local variable and member fields, local variable takes precedence. Formal parameters are located after local variable and are the first one which the compiler declares. Hence, local variables cannot named be named as formal parameters.

- 21 Do answer directed questions
- Q. Define Variable, Access modifier
- A variable is a container that holds value that are used in a java program.
- As the name suggests access modifiers in java helps to restrict the scope of a variable.

class, constructor, variable, method or data member.

B. Enlist types of variable.

There are three types of variable in Java

1. Local

2. Instance

3. Reference static

C. Difference between static and instance variable:

Instance variable : one copy per object. Every object has its own instance.

Ex. x, y, r (center and radius in the circle)

(static variable: one copy per class.)

Ex. numCircle (total number of circle objects created)

22 Do as directed & write output

1. $9 \gg 3 \rightarrow 1$

2. $8 \ll 3 \rightarrow 64$

3. $-12 \gg 4 \rightarrow -7$

4. $-60 \ll 4 \rightarrow -960$

23 If a constructor's parameter name and member variable name and member variable name are same then, how can one can distinguish both variables? Write an example for same.

→ This keyword is used to distinguish between variable name and constructor's parameter name.

Student's Signature

Teacher's Signature

Example

```
class Account {
```

 int a;
 int b;

```
    Account() {  
        a = 100;  
        b = 200;  
    }
```

```
    Account(int a, int b) {  
        this.a = a;  
        this.b = b;  
    }
```

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

```
        Account obj = new Account(2, 2);
```

 System.out.println("a = " + obj.a + " b = " + obj.b);

3

- 13 Write a statement to declare and instantiate an array to hold marks obtained by students in different subjects in a class. Assume that there are up to 30 students in a class and there are 5 subjects.

```
→ class Student {
```

```
    public static void main(String args) {
```

```
        int array[] = new int[30][5];
```

3

3

example 2 solution

20 State the usage of getter and setter method using practical example.

→ In java, getter and setters are two conventional methods that are used for retrieving and updating value of a variable.

Example :

```
public class Get_Set {
    private int num;
    public int getNum() {
        return this.num;
    }
}
```

```
public void setNum(int num) {
    this.num = num;
}
```

The class declares a private variable, num. Since num is private, code from outside this class cannot access the variable directly. Like :

Example :
Get_Set obj = new Get_Set();
obj.num = 70; // Compile error, private variable
int number = obj.num; // error

Assignment

Teacher's Signature

Date _____
Page _____

Outside code have to invoke the getter, getnum() and the setter setnum() in order to read or update the variable.

Example :

```
creat set obj = new creat set();
obj.setnum(10); // Done
int number = obj.getnum(); // Done
```