FULL STACK DEVELOPMENT – WORKSHEET 3 Answers

1. (B) Explanation: Pointers are not a Java feature. Java provides an efficient abstraction layer for developing without using a pointer in Java. Features of Java Programming are Portable, Architectural Neutral, Object-Oriented, Robust, Secure, Dynamic and Extensible, etc.

2. (C) Explanation: Keywords are specially reserved words that can not be used for naming a user-defined variable, for example: class, int, for, etc.

3. (C) Explanation: Object class is a superclass of every class in Java.

4. (C) Explanation: Boolean can only be assigned true or false literals.

5. (D) Explanation: Default is the access modifier when none is defined explicitly. It means the member (method or variable) can be accessed within the same package.

6. (C) Explanation: Variables of an interface are public, static and final by default because the interfaces cannot be instantiated, final ensures the value assigned cannot be changed with the implementing class and public for it to be accessible by all the implementing classes.

7. (C)

8. (A) Infinite

9. (C) This

10.(B) 3 Explanation: The Math.random() method returns a number greater than or equal to 0 and less than 1. so 2.5 will be greater than or equal to 2.5 and less than 3.5, we can be sure that Math.round() will round that number to 3.

11. (D) Explanation: max(), min() and abs() are all rounding functions.

12.(D) Explanation: Standard output variable ‘out’ is defined in System class. out is usually used in print statement i:e System.out.print().

13.(C) Explanation: Two variables with the same name can’t be created in a class.

14.(B) Explanation: If a static method is present in the program then it will be executed first, then main will be executed.

15.(D)

16.(D)Explanation: main() method must be made public. Without main() being a public java run time system will not be able to access main() and will not be able to execute the code.

17.Error.It will not calculate volume.To return volume do the following:change method return type from void to int.return volume from method.
define variables in main to catch with return type.
void volume() . int volume(){int volume= l*b*h;return volume}
obj.volume(); . int x = obj.volume(); print(x)

18.Syntax:

```
public class Main {
  static void myMethod() {
    System.out.println("Hello All");
  }

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

Syntax:Declare a method

<access_modifier> <return_type> <method_name>(list_of_parameters)

{

//body

```
}
```

19. // Java Program to Illustrate Methods and Method call

```
package com.java.worksheet3;
import java.util.Scanner;
	class Sum
	{
		int c;
		void addition(int x,int y)
		{
		 c=x+y;
		}
	public static void main(String[] arg)
	{
	int a,b;
	Scanner sc=new Scanner(System.in);
	System.out.println("Enter first number");
	a=sc.nextInt();
	System.out.println("Enter second number");
	b=sc.nextInt();
	Sum r=new Sum();
	r.addition(a,b);
	System.out.println("Addition of two numbers is : "+r.c);
	}
	}
```

20.// Java Program to use getter and setter methods

```
// Define a class
```

```
class Example {
// Define instance variables
private int number;
private String name;
// Define accessor (getter) methods
public int getNumber() {
return number;
}
public String getName() {
return name;
}
// Define mutator (setter) methods
public void setNumber(int number) {
this.number = number;
}
public void setName(String name) {
this.name = name;
}
// Define other methods
public void printDetails() {
System.out.println("Number: " + number);
System.out.println("Name: " + name);
}
public class Demo{
public static void main(String[] args) {
Example example = new Example();
example.setNumber(123);
```

```
example.setName("Your Name");

example.printDetails();

} }

}
```