LAB ASSIGNMENT (21 August)

1. **Snippet 1:**
   a. **Error Identification: -** Main method is not static in class Main. The 'static' was missing in public static void main statement.
   b. **Brief Explanation:** Basically the *main()* method is static so that JVM can invoke it without **instantiating the class**. This also saves the unnecessary wastage of memory which would have been used by the object declared only for calling the *main()* method by the JVM.
   c. **Fixed Errors :-**

```
public class Main {

public static void main(String[] args) {
System.out.println("Hello, World!");
}
}
```

2. **Snippet 2:**
   a. **Error Identification**: 'public' missing before static void main statement.
   b. **Brief Explanation:** It is an Access modifier, which specifies from where and who can access the method. Making the main() method public makes it globally available. It is made public so that JVM can invoke it from outside the class as it is not present in the current class.
   c. **Fixed Errors :-** public class Main {

```
 public static void main(String[] args) {
 System.out.println("Hello, World!");
 }
}
```

Q. What happens when you compile and run this code?

➔ ERROR!

Error: Main method not found in class Main, please define the main method as:

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

or a JavaFX application class must extend javafx.application.Application

3. Snippet 3:-

Q. What error do you encounter? Why is void used in the main method?

➔ ERROR!

Error: Main method must return a value of type void in class Main, please define the main method as:

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

Void is a keyword and is used to specify that a method doesn't return anything. As the *main()* method doesn't return anything, its return type is void. As soon as the *main()* method terminates, the Java program terminates too.

Fixed Errors :- public class Main {

```
 public static void main(String[] args) {
 System.out.println("Hello, World!");
 }
}
```

4. Snippet 4:

   Q. What happens when you compile and run this code? Why is String[] args needed?

   ➔ ERROR!

   Error: Main method not found in class Main, please define the main method as:

   public static void main(String[] args)

   or a JavaFX application class must extend javafx.application.Application

   String[] args is needed in the main method because it allows your program to accept arguments (inputs) from the command line when you run it.

   Corrected Code:-

```
public class Main {
 public static void main(String[] args) {
 System.out.println("Hello, World!");
 }
}
```

5. **Can you have multiple main methods? What do you observe?**

   ➔ Yes we can have multiple main methods . The code executed successfully but gave only one print statement output.

6. **What error occurs? Why must variables be declared?**

   ⇨ **int x = y + 10;**

   ⇨ **^**

   ⇨ **symbol: variable y**

   ⇨ **location: class Main**

   ⇨ **1 error**

   Every variable is assigned a data type that designates the type and quantity of value it can hold. Variable is a memory location name of the data. A variable is a name given to a memory location that's why it must be declared

7. What compilation error do you see? Why does Java enforce type safety?

   ➔ error: incompatible types: String cannot be converted to int

   int x = "Hello";

   It enforces type safety because it prevents errors and makes sure that operations are performed only with compatible types. Also allows to catch the errors during

compilation before running the code. Also ensures that memory manipulates correctly.

Corrected code:

```
public class Main {
 public static void main(String[] args) {
 String x = "Hello";
 System.out.println(x);
 }
}
```

8. What syntax errors are present? How do they affect compilation?

   ⇨ The closing bracket of print statement and semi-colon is missing. It throws the error that ‘)’ is missing.

9. What error occurs? Why can't reserved keywords be used as identifiers?

   ⇨ <identifier> expected

```
int class = 10;
        ^
```

ERROR!

/tmp/FwqFbc73De/Main.java:4: error: <identifier> expected

We can’t use reserved keywords because they are already predefined in syntax of java. It is not allowed to use because it can create errors and confusion for compiler as

well as the developers. It can create compiler's job difficult to differentiate between them.

10. What happens when you compile and run this code? Is method overloading allowed?

➔ It throws error : non-static method display() cannot be referenced from a static context. The method overloading is allowed in Java.

11. What runtime exception do you encounter? Why does it occur?

⇨ ava.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 3

⇨ This error occurs because we are trying to access the the index 5 and array is of size 3 only

12. What happens when you run this code? How can you avoid infinite loops?

➔ When we run this code it goes to infinite loop and prints that infinite loop statement for infinite times. We should ensure loops are correctly defined and can be evaluated to false. Proper use of break statements can be done. Testing and debugging can be done

13. What exception is thrown? Why does it occur?

⇨ Exception in thread "main"
java.lang.NullPointerException: Cannot invoke "String.length()" because "<local1>" is null
at Main.main(Main.java:4)

14. What compilation error occurs? Why does Java enforce data type constraints?

⇨ error: incompatible types: String cannot be converted to double
Constraints help the developers to define the roles and conditions that must be met for their code to function correctly. They also ensure data integrity and help to protect the unexpected behaviour of the program.

15. What error occurs when compiling this code? How should you handle different data types in operations?

⇨ error: incompatible types: possible lossy conversion from double to int. We must ensure that data types in an operation are compatible. We can type cast that is convert them intentionally . Try to keep data operations of the same data type.

16. What is the result of this operation? Is the output what you expected?

⇨ Result of this operation is 2.0 as it is using double. For exact answer that is 2.5 we have to use (double) before num/4 to typecast it

17. What compilation error occurs? Why is the ** operator not valid in Java?

⇨ error: illegal start of expression
int result = a ** b;
The ** operator is not valid in Java because the language does not include it as part of its syntax. Instead, Java uses the Math.pow() method for performing exponentiation

18. What is the output of this code? How does operator precedence affect the result?

⇨ Output will be 20. Java follows the BODMAS rule that's why multiplication done first then addition is done.

19. What runtime exception is thrown? Why does division by zero cause an issue in Java?

⇨ ERROR!
Exception in thread "main"
java.lang.ArithmeticException: / by zero
This error is occurred because we cannot divide anything by zero. Something divided by 0 is Infinity.

20. What syntax error occurs? How does the missing semicolon affect compilation?

⇨ ;' expected

System.out.println("Hello, World")

^1 error

Semicolon is needed after each statement completion. It is used to terminate the statement.

21. What does the compiler say about mismatched braces?

⇨ eached end of file while parsing }

the brackets are not completed because that curly bracket is opened but not closed.

22. What syntax error occurs? Can a method be declared inside another method?

illegal start of expression

static void displayMessage() {

^

ERROR!

/tmp/jQ0DTC9Qzz/Main.java:7: error: class, interface, or enum expected

}

^

2 errors Method Cannot be declared inside a method

23. Error to Investigate: Why does the default case print after "Value is 2"? How can you prevent the program from executing the default case?

⇨ Default case is printed because we have not used the break. We can use the break statement to prevent the program from executing the default case.

24. When level is 1, why does it print "Level 1", "Level 2", "Level 3", and "Unknown level"? What is the role of the break statement in this situation?

⇨ Level 1 is the first S.O.P statement and it prints all the below statements as the break is not used. The role of break statement is to exit the program when the level 1 is reached.

25. Why does this code not compile? What does the error tell you about the types allowed in switch expressions? How can you modify the code to make it work?

⇨ incompatible types: possible lossy conversion from double to int

switch(score) {

It is because we have declared the scores as doble and in cases we have used integer. We can declare the score as int or we can use cases as double like 85.0, 100.0

26. Why does the compiler complain about duplicate case labels? What happens when you have two identical case labels in the same switch block?

⇨ The interpreter does not check that these expressions are all different. At runtime, if two cases in a switch statement have the same label, the second case will never be executed.