Assignment 2

1.What error do you get when running this code?

‘main' method is not declared 'public static

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

Main method not found in class Main, please define the main method as:
public static void main(String[] args)

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

Main method not found in class Main, please define the main method as:
public static void main(String[] args)

main method ends at the end of the program and hence it doesn't return anything

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

Main method not found in class Main, please define the main method as:
public static void main(String[] args)

String args[] are required for accepting command line arguments

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

we can have multiple main method but the compiler will compile only main method with the String args[]

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

cannot find symbol
int x = y + 10;

^
symbol: variable y
location: class Main

variable are declared so that the compiler will know the data type and work with it accordingly

it also allows us to allocate memory for the variable

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

Main.java:3: error: incompatible types: String cannot be converted to int
int x = "Hello";
^
1 error

java enforce type safety to check whether the datatypes and value inside the variable match and their is no need of explicit type casting and any runtime error

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

error: ')' or ',' expected
System.out.println("Hello, World!"

at compile time if the compiler gets any syntax error it will not execute the code and will throw an error

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

not a statement
int class = 10;
^
Main.java:3: error: ';' expected
int class = 10;
^
Main.java:3: error: <identifier> expected
int class = 10;

^
Main.java:4: error: illegal start of expression
System.out.println(class);
^
Main.java:4: error: <identifier> expected
System.out.println(class);
^

reserved keyword have predefined meaning hence we use them as an identifiers .

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

non-static method display() cannot be referenced from a static context
display();
^
Main.java:10: error: non-static method display(int) cannot be referenced from a static context
display(5);
^

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

Index 5 out of bounds for length 3
at Main.main(Main.java:4)

index out of bound error occurs when we try to access index that is not declared or doesn't exist

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

it goes inside infinite loop and never stops

to avoid infinite loop we should provide an condition that will terminate after some time

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)

because we are checking the length of an String that is pointing to Null

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

incompatible types: String cannot be converted to double
double num = "Hello";

java enforce type constraints to avoid data mismatch and to make sure data stored in the type are consistent

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

for handling this we can typecast it with int

incompatible types: possible lossy conversion from double to int
int result = num1 + num2;

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

2.0

yes the output is as per expected

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

illegal start of expression
int result = a ** b;
it is not valid operator valid operator for multiplication is *

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

output 20;

it follows the bodmas rule it will first multiply and the add the product and this will provide slightly different output

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

Exception in thread "main" java.lang.ArithmeticException: / by zero
at Main.main(Main.java:5)

the operator precedence follows bodmas rule that means if the higher priority operator is present in the equation then it will operate on them first and then will move as per

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

the compiler will not know whether the statement is over or not

ain.java:3: error: ';' expected
System.out.println("Hello, World")
^
1 error

21.What does the compiler say about mismatched braces?

reached end of file while parsing
}

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

Main.java:3: error: illegal start of expression
static void displayMessage() {
^
Main.java:7: error: class, interface, enum, or record expected
}

^

2 errors

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

to prevent the programme from executing default we have to add break to the end of each case or second 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?

it prints all the cases because we haven't used any break statement here

the break statement is used to declare the scope of the case

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?

elector type double is not allowed
switch(score) {

floats are not exact that means two different numbers can be assumed as same that means the code will be meaningless

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 cases should not be duplicate or else it will create an ambiguity issue