# Assignment 2

## Section 1: Error-Driven Learning in Java

### Snippet 1:

public class Main {

public void main(String[] args) {

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

}

}

Q1. What error do you get when running this code?

Ans.1 Main method is not static in class Main, please define the main method as: public static void main (String[] args). This is a compilation error.

Q2. Write a brief explanation of error and its cause.

Ans.2 Static keyword is missing. Without the static keyword, the method cannot be called without creating an object.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

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

}

}

Q4. Explanation of Snippet.

Ans.4 The code will print Hello, World!.

### Snippet 2:

public class Main {

static void main(String[] args) {

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

}

}

Q1. What error do you get when running this code?

Ans.1 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. This is a runtime error.

Q2. Write a brief explanation of error and its cause.

Ans.2 Public keyword is missing. Without the public keyword, the method is not allowed to be called from outside the class by JRE.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

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

}

}

Q4. Explanation of Snippet.

Ans.4 The code will print Hello, World!.

### Snippet 3:

public class Main {

public static int main(String[] args) {

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

return 0;

}

}

Q1. What error do you get when running this code?

Ans.1 Main method must return a value of type void in class Main, please define the main method as: public static void main(String[] args). This is a runtime error.

Q2. Write a brief explanation of error and its cause.

Ans.2 In this snippet, int is the return type of the main method. The main method does not return anything after it executes. Therefore, the int should be replaced by void.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

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

}

}

Q4. Explanation of Snippet.

Ans.4 The code will print Hello, World!.

### Snippet 4:

public class Main {

public static void main() {

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

}

}

Q1. What error do you get when running this code?

Ans.1 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. This is a runtime error.

Q2. Write a brief explanation of error and its cause.

Ans.2 String[] args keyword is missing inside the main method parenthesis. These are the string of arrays that are going to be passed in the program. This is a provision given by Java to pass arguments in this string array. But it is not optional and has to be written to run the program.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

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

}

}

Q4. Explanation of Snippet.

Ans.4 The code will print Hello, World!.

### Snippet 5:

public class Main {

public static void main(String[] args) {

System.out.println("Main method with String[] args");

}

public static void main(int[] args) {

System.out.println("Overloaded main method with int[] args");

}

}

Q1. Can you have multiple main methods? What do you observe?

Ans.1 Yes, there can be multiple main methods in a program but the output will only be displayed of the one with string of arrays as argument in main method. If we comment the main method with string of arrays as argument, the program will give the following 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. This is a runtime error.

### Snippet 6:

public class Main {

public static void main(String[] args) {

int x = y + 10;

System.out.println(x);

}

}

Q1. What error do you get when running this code?

Ans.1 cannot find symbol int x = y + 10;

^

symbol: variable y

location: class Main. This is a compile time error.

Q2. Write a brief explanation of error and its cause.

Ans.2 The variable y has not been declared and initialized in the program. Declaring a variable will inform the compiler about the type of data the variable will hold and how much memory to allocate for it. Initializing a variable will assign a value to it after which the addition operation can be performed.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

int y = 0;

int x = y + 10;

System.out.println(x);

}

}

Q4. Explanation of Snippet.

Ans.4 The variable y is declared as integer and has been assigned a zero value. Addition of value in y and 10 has been performed which is assigned to x variable. Then the output of x is printed.

### Snippet 7:

public class Main {

public static void main(String[] args) {

int x = "Hello";

System.out.println(x);

}

}

Q1. What error do you get when running this code?

Ans.1 incompatible types: String cannot be converted to int

int x = “Hello”;

^

Q2. Write a brief explanation of error and its cause.

Ans.2 The error is caused because the variable is of integer type and the value assigned to it is a string.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

String x = "Hello";

System.out.println(x);

}

}

Q4. Explanation of Snippet.

Ans.4 The program will print “Hello”.

Q5. Why does Java enforce type safety?

Ans.5 Java prevents the programs from accessing memory in inappropriate ways by controlling the memory access of each object.

### Snippet 8:

public class Main {

public static void main(String[] args) {

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

}

}

Q1. What error do you get when running this code?

Ans.1 error: ')' or ',' expected

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

^

Q2. Write a brief explanation of error and its cause.

Ans.2 This is caused by a syntactical error. The closing bracket and the semi-colon is missing after the Hello, World! statement.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

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

}

}

Q4. Explanation of Snippet.

Ans.4 The code will print Hello, World!.

### Snippet 9:

public class Main {

public static void main(String[] args) {

int class = 10;

System.out.println(class);

}

}

Q1. What error do you get when running this code?

Ans.1 Main.java:3: error: 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);

^

Q2. Write a brief explanation of error and its cause.

Ans.2 Reserved word class is used in place of an identifier. Reserved words have predefined meanings in the language’s syntax and using them as identifiers would lead to confusion and syntax errors.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

int x = 10;

System.out.println(x);

}

}

Q4. Explanation of Snippet.

Ans.4 The code will print the value 10 assigned to x.

### Snippet 10:

public class Main {

public void display()

{

System.out.println("No parameters");

}

public void display(int num)

{

System.out.println("With parameter: " + num);

}

public static void main(String[] args)

{

display();

display(5);

}

}

Q1. What error do you get when running this code?

Ans.1 Main.java:12: error: non-static method display() cannot be referenced from a static context

display();

^

Main.java:13: error: non-static method display(int) cannot be referenced from a static context

display(5);

^

Q2. Write a brief explanation of error and its cause.

Ans.2 While declaring both the methods, static keyword is missing. Without the static keyword, the method cannot be called without creating an object.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void display()

{

System.out.println("No parameters");

}

public static void display(int num)

{

System.out.println("With parameter: " + num);

}

public static void main(String[] args)

{

display();

display(5);

}

}

Q4. Explanation of Snippet.

Ans.4 display() will give No parameters as output while display(5) will give With parameter : 5 as output.

Q5. Is method overloading allowed?

Ans.5 Yes.

### Snippet 11:

public class Main {

public static void main(String[] args) {

int[] arr = {1, 2, 3};

System.out.println(arr[5]);

}

}

Q1. What error do you get when running this code?

Ans.1 Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 3

at Main.main(Main.java:4)

Q2. Write a brief explanation of error and its cause.

Ans.2 The error occurred because the value that can be passed to string array is either 0, 1 or 2 as there are only three values in array.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

int[] arr = {1, 2, 3};

System.out.println(arr[2]);

}

}

Q4. Explanation of Snippet.

Ans.4 The program will print the respective value of array as per the index passed in print statement.

### Snippet 12:

public class Main {

public static void main(String[] args) {

while (true)

{

System.out.println("Infinite Loop");

}

}

}

Q1. What happens when you run this code? How can you avoid infinite loops?

Ans.1 The program will enter into an infinite loop. To avoid this, the condition in the while loop should be such that it becomes false at some point so that the compiler can come out of loop.

### Snippet 13:

public class Main {

public static void main(String[] args) {

String str = null;

System.out.println(str.length());

}

}

Q1. What error do you get when running this code?

Ans.1 Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.length()" because "<local1>" is null

at Main.main(Main.java:4). This is a compilation error.

Q2. Write a brief explanation of error and its cause.

Ans.2 The error occurred because null value has been assigned to string and length() cannot find the length of null value.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

String str = "Dipesh";

System.out.println(str.length());

}

}

Q4. Explanation of Snippet.

Ans.4 The program will print the length of the string passed in str variable.

### Snippet 14:

public class Main {

public static void main(String[] args) {

double num = "Hello";

System.out.println(num);

}

}

Q1. What error do you get when running this code?

Ans.1 Main.java:3: error: incompatible types: String cannot be converted to double

double num = "Hello";

^

Q2. Write a brief explanation of error and its cause.

Ans.2 The error is caused because the variable is of double type and the value assigned to it is a string.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

String x = "Hello";

System.out.println(x);

}

}

Q4. Explanation of Snippet.

Ans.4 The program will print “Hello”.

Q5. Why does Java enforce data type constraints?

Ans.5 Java prevents the programs from accessing memory in inappropriate ways by controlling the memory access of each object.

### Snippet 15:

public class Main {

public static void main(String[] args) {

int num1 = 10;

double num2 = 5.5;

int result = num1 + num2;

System.out.println(result);

}

}

Q1. What error do you get when running this code?

Ans.1 Main.java:5: error: incompatible types: possible lossy conversion from double to int

int result = num1 + num2;

Q2. Write a brief explanation of error and its cause.

Ans.2 The error occurred because the data type that are being added are different. Explicit type conversion has to be performed to resolve the error.

Q3. Modify the code to correct the error.

Ans.3 public class Main {

public static void main(String[] args) {

int num1 = 10;

double num2 = 5.5;

int result = num1 + (int)num2;

System.out.println(result);

}

}

Q4. Explanation of Snippet.

Ans.4 Program will add the numbers in num1 and num2 variables and displays its result as output.

### Snippet 16:

public class Main {

public static void main(String[] args) {

int num = 10;

double result = num / 4;

System.out.println(result);

}

}

Q1. What is the result of this operation? Is the output what you expected?

Ans.1 The output of the program is 2.0. It is because both 10 and 4 are integers so the result of division is 2. Since we are storing this value into a variable of type double, therefore .0 gets added into it.

### Snippet 17:

public class Main {

public static void main(String[] args) {

int a = 10;

int b = 5;

int result = a \*\* b;

System.out.println(result);

}

}

Q1. What error do you get when running this code?

Ans.1 Main.java:5: error: illegal start of expression

int result = a \*\* b;

Q2. Write a brief explanation of error and its cause.

Ans.2 \*\* operator is not a valid operator in java to calculate power.

Q3. Modify the code to correct the error.

And.3 public class Main {

public static void main(String[] args) {

int a = 10;

int b = 5;

//int result = Math.pow(a,b);

System.out.println(Math.pow(a,b));

}

}

Q4. Explanation of Snippet.

Ans.4 This program returns 10 to the power of 5.

### Snippet 18:

public class Main {

public static void main(String[] args) {

int a = 10;

int b = 5;

int result = a + b \* 2;

System.out.println(result);

}

}

Q1. What is the output of this code? How does operator precedence affect the result?

Ans.1 The output of this code is 20. As per the precedence rule, the multiplication operation will carry out first and the output of multiplication will be added with a.

### Snippet 19:

public class Main {

public static void main(String[] args) {

int a = 10;

int b = 0;

int result = a / b;

System.out.println(result);

}

}

Q1 What runtime exception is thrown? Why does division by zero cause an issue in Java?

Ans.1 Exception in thread "main" java.lang.ArithmeticException: / by zero

at Main.main(Main.java:5). Division by zero causes an issue in Java because any number divided by zero is undefined in mathematics.

### Snippet 20:

public class Main {

public static void main(String[] args) {

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

}

}

Q1. What syntax error occurs? How does the missing semicolon affect compilation?

Ans.1 Main.java:3: error: ';' expected

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

^

Semicolon tells the compiler where an instruction ends and where the next instruction begins.

### Snippet 21:

public class Main {

public static void main(String[] args) {

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

// Missing closing brace here

}

Q1. What does the compiler say about mismatched braces?

Ans.1 Main.java:5: error: reached end of file while parsing

}

^

1 error

### Snippet 22:

public class Main {

public static void main(String[] args) {

static void displayMessage() {

System.out.println("Message");

}

}

}

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

Ans. Main.java:3: error: illegal start of expression

static void displayMessage() {

^

Main.java:7: error: class, interface, enum, or record expected

}

^

2 errors.

You can’t declare a metho directly within a method but you can declare a whole class within the scope of a method and that class can have as many methods as it likes.

### Snippet 23:

public class Confusion {

public static void main(String[] args) {

int value = 2;

switch(value) {

case 1:

System.out.println("Value is 1");

case 2:

System.out.println("Value is 2");

case 3:

System.out.println("Value is 3");

default:

System.out.println("Default case");

}

}

}

Q. Why does the default case print after "Value is 2"? How can you prevent

the program from executing the default case?

Ans. Because break token is missing after each case. By adding break token after each case statements.

### Snippet 24:

public class MissingBreakCase {

public static void main(String[] args) {

int level = 1;

switch(level) {

case 1:

System.out.println("Level 1");

case 2:

System.out.println("Level 2");

case 3:

System.out.println("Level 3");

default:

System.out.println("Unknown level");

}

}

}

Q. 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?

Ans. Because break tokens are missing after every case statements. When the condition satisfies, the compiler reads the break statement and does not go further to read other cases.

### Snippet 25:

public class Switch {

public static void main(String[] args) {

double score = 85.0;

switch(score) {

case 100:

System.out.println("Perfect score!");

break;

case 85:

System.out.println("Great job!");

break;

default:

System.out.println("Keep trying!");

}

}

}

Error to Investigate: 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?

Ans. Main.java:4: error: selector type double is not allowed

switch(score) {

^

1 error

The error tells that double type expressions are not allowed in switch statements. There are only four types of data types allowed in switch case statements. They are char, short, int, and byte. Replace the double by byte in the code.

### Snippet 26:

public class Switch {

public static void main(String[] args) {

int number = 5;

switch(number) {

case 5:

System.out.println("Number is 5");

break;

case 5:

System.out.println("This is another case 5");

break;

default:

System.out.println("This is the default case");

}

}

}

Error to Investigate: Why does the compiler complain about duplicate case labels? What

happens when you have two identical case labels in the same switch block?

Ans. Main.java:8: error: duplicate case label

case 5:

^

The compiler is complaining about same values have been passed in both the cases due to which it is unable to decide the values to print.

## Section 2: Java Programming with Conditional Statements

Question 1: Grade Classification

Write a program to classify student grades based on the following criteria:

 If the score is greater than or equal to 90, print "A"

 If the score is between 80 and 89, print "B"

 If the score is between 70 and 79, print "C"

 If the score is between 60 and 69, print "D"

 If the score is less than 60, print "F"

Ans.1 class Grade {

public static void main(String[] args) {

int score = 85;

if (score >= 90)

{

System.out.println("A");

}

else if (score>=80 && score <=89)

{

System.out.println("B");

}

else if (score>=70 && score <=79)

{

System.out.println("C");

}

else if (score>=60 && score <=69)

{

System.out.println("D");

}

else

{

System.out.println("F");

}

}

}

Question 2: Days of the Week

Write a program that uses a nested switch statement to print out the day of the week based on an

integer input (1 for Monday, 2 for Tuesday, etc.). Additionally, within each day, print whether it

is a weekday or weekend.

class Days {

public static void main(String[] args) {

int day = 1;

switch(day)

{

case 1:

{

System.out.println("Monday is a weekday");

break;

}

case 2:

{

System.out.println("Tuesday is a weekday");

break;

}

case 3:

{

System.out.println("Wednesday is a weekday");

break;

}

case 4:

{

System.out.println("Thursday is a weekday");

break;

}

case 5:

{

System.out.println("Friday is a weekday");

break;

}

case 6:

{

System.out.println("Saturday is a weekend");

break;

}

case 7:

{

System.out.println("Sunday is a weekend");

break;

}

default:

{

System.out.println("Invalid input");

break;

}

}

}

}

Question 3: Calculator

Write a program that acts as a simple calculator. It should accept two numbers and an operator

(+, -, \*, /) as input. Use a switch statement to perform the appropriate operation. Use nested ifelse

to check if division by zero is attempted and display an error message.

import java.util.Scanner;

class Calculator {

public static void main(String[] args){

Scanner myObj = new Scanner(System.in);

System.out.println("Enter a number");

int n1 = myObj.nextInt();

Scanner myObj1 = new Scanner(System.in);

System.out.println("Enter a number");

int n2 = myObj1.nextInt();

Scanner myObj2 = new Scanner(System.in);

System.out.println("Enter an operator");

char operator = myObj2.next().charAt(0);

switch(operator)

{

case '+':

{

System.out.println("Sum of " + n1 + " and " + n2 + " is: " + (n1+n2));

break;

}

case '-':

{

System.out.println("Subtraction of " + n2 + " from " + n1 + " is: " + (n1-n2));

break;

}

case '\*':

{

System.out.println("Product of " + n1 + " and " + n2 + " is: " + n1\*n2);

break;

}

case '/':

{

if(n2==0)

{

System.out.println("Division by 0 is not possible.");

break;

}

else

{

System.out.println("Quotient of " + n1 + " and " + n2 + " is: " + n1/n2);

break;

}

}

default:

{

System.out.println("Invalid input");

break;

}

}

}

}

Question 4: Discount Calculation

Write a program to calculate the discount based on the total purchase amount. Use the following

criteria:

 If the total purchase is greater than or equal to Rs.1000, apply a 20% discount.

 If the total purchase is between Rs.500 and Rs.999, apply a 10% discount.

 If the total purchase is less than Rs.500, apply a 5% discount.

Additionally, if the user has a membership card, increase the discount by 5%.

import java.util.Scanner;

class Discount {

public static void main(String[] args){

Scanner myObj = new Scanner(System.in);

System.out.println("Do you have membership card?");

char membershipcard = myObj.next().charAt(0);

Scanner myObj1 = new Scanner(System.in);

System.out.println("Enter total purchase amount");

double tpa = myObj1.nextInt();

if (tpa>=1000)

{

if (membershipcard == 'Y')

{

tpa = tpa\*0.75;

}

else

{

tpa = tpa\*0.8;

}

}

else if (tpa>=500 && tpa<=999)

{

if (membershipcard == 'Y')

{

tpa = tpa\*0.85;

}

else

{

tpa = tpa\*0.9;

}

}

else

{

if (membershipcard == 'Y')

{

tpa = tpa\*0.9;

}

else

{

tpa = tpa\*0.95;

}

}

System.out.println("Total Purchase Amount after discount = " + tpa);

}

}

Question 5: Student Pass/Fail Status with Nested Switch

Write a program that determines whether a student passes or fails based on their grades in three

subjects. If the student scores more than 40 in all subjects, they pass. If the student fails in one or

more subjects, print the number of subjects they failed in.

import java.util.Scanner;

class Status {

public static void main(String[] args){

Scanner myObj1 = new Scanner(System.in);

System.out.println("Enter marks of S1");

int m1 = myObj1.nextInt();

Scanner myObj2 = new Scanner(System.in);

System.out.println("Enter marks of S2");

int m2 = myObj2.nextInt();

Scanner myObj3 = new Scanner(System.in);

System.out.println("Enter marks of S3");

int m3 = myObj3.nextInt();

int failcount = 0;

switch(m1>40?1:0)

{

case 0:

{

failcount++;

break;

}

}

switch(m2>40?1:0)

{

case 0:

{

failcount++;

break;

}

}

switch(m3>40?1:0)

{

case 0:

{

failcount++;

break;

}

}

switch(failcount)

{

case 0:

{

System.out.println("Student has passed");

break;

}

default:

{

System.out.println("Student has failed in "+ failcount + " subject(s)");

break;

}

}

}

}