**Assignment 2**

**Lab Assignment**

**Snippet 1:**

public class Main {

public void main(String[] args) {

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

}

}

D:\Feb 25>javac Main.java

D:\Feb 25>java Main

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

public static void main(String[] args)

**Corrected Code**

public class Main {

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

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

}

}

**Output:**

D:\Feb 25>javac Main.java

D:\Feb 25>java Main

Hello, World!

**Snippet 2:**

public class Main {

static void main(String[] args) {

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

}

}

**Error**

D:\Feb 25>javac Main.java

D:\Feb 25>java Main

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

**Corrected Code**

public class Main {

public static void main(String[] args) {

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

}

}

D:\Feb 25>javac Main.java

D:\Feb 25>java Main

Hello, World!

=================================================================

***Snippet 3:***

public class Main {

public static int main(String[] args) {

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

Return 0;

} }

**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)

Main.java:5: error: incompatible types: unexpected return value

return 0;

^

**Corrected Code:**

public class Main {

public static void main(String[] args) {

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

}

}

=======================================================

***Snippet 4:***

public class Main {

public static void main() {

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

}

}

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

**Corrected Code:**

public class Main {

public static void main(String[] args) {

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

}

}

String[] args is needed cause it is an array of strings that stores command- line arguments passed to the program.

==================================================================

***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");

}

}

**There is no error.**

**Yes we have a multiple main methods but it executes the first one.**

**Output:**

Main method with String[] args

===================================================================

***Snippet 6:***

public class Main {

public static void main(String[] args) {

int x = y + 10;

System.out.println(x);

}

}

**Error:**

Main.java:4: error: cannot find symbol

int x = y + 10;

^

symbol: variable y

location: class Main

**Error occurs because we have not declared the variable. It is requires to stores input values as a reference to used in program.**

***==================================================================***

***Snippet 7:***

public class Main {

public static void main(String[] args) {

int x = "Hello";

System.out.println(x);

}

}

**Error:**

**Main.java:4: error: incompatible types: String cannot be converted to int**

**int x = "Hello";**

**Because here we can’t assign a string value to int.**

**Java enforce type safety cause**

1. **It prevents Runtime error**
2. **Enables compile time error detection**
3. **Improves code readability**

**Correcfed code:**

public class Main {

public static void main(String[] args) {

int x = 17;

System.out.println(x);

}

}

**Output:**

**17**

**================================================================**

***Snippet 8:***

public class Main {

public static void main(String[] args) {

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

}

}

**Error:**

Main.java:5: error: ')' expected

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

Compilation stops at the line where )is missing. Program will not work.

**Output:**

Hello, World!

====================================================================

***Snippet 9:***

public class Main {

public static void main(String[] args) {

int class = 10;

System.out.println(class);

}

}

 **What error occurs? Why can't reserved keywords be used as identifiers?**

Errors:

illegal start of type

<identifier> expected

Corrected code:

public class Main {

public static void main(String[] args) {

int a = 10;

System.out.println(a);

}

}

**Output:**

10

We can’t use reserved keywords as a identifiers cause it has a predefined meaning.

====================================================================

***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);

}

}

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

**Error:**

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

display();

^

Main.java:11: error: non-static method display(int) cannot be referenced from a static context

display(5);

**Corrected code;**

==================================================================

***Snippet 11:***

public class Main {

public static void main(String[] args) {

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

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

}

}

 What runtime exception do you encounter? Why does it occur?

**Errors:**

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

at Main.main(Main.java:5)

This exception is showing becausing in our array 2 is 0, 1,2

And here we trying to access 5 .

=====================================================================

***S****nippet 12:*

public class Main {

public static void main(String[] args) {

while (true) {

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

}

}

}

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

After running the code it will run infinitely.

We can avoid infinite loops by giving a condition.

**Corrected code:**

public class Main {

public static void main(String[] args) {

int a=10;

while (a>=5) {

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

a--;

}

}

}

Output:

Infinite Loop

Infinite Loop

Infinite Loop

Infinite Loop

Infinite Loop

Infinite Loop

================================================================

***Snippet 13:***

public class Main {

public static void main(String[] args) {

String str = null;

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

}

}

 What exception is thrown? Why does it occur?

Error:

Exception in thread "main" java.lang.NullPointerException

at Main.main(Main.java:5)

**it occurs when we try to access the value which assign to null.**

**Corrected code:**

public class Main {

public static void main(String[] args) {

String str = "Harshada";

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

}

}

**Output:**

**8**

**=================================================================**

***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);

}

}

**Error:**Main.java:6: error: incompatible types: possible lossy conversion from double to int

int result = num1 + num2;

^

Here num1 and num2 are different data types and we are trying add them store them in int data type. It is not possible cause double has higher precision then int.

We can handle different data types by using type casting. But in result there is loss of decimal precision.

**Corrected code:**

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);

}

}

**Output:**

15

**=================================================================**

***Snippet 16:***

public class Main {

public static void main(String[] args) {

int num = 10;

double result = num / 4;

System.out.println(result);

}

}

output:

2.0

Yes this output is expected cause data type result is double and double has higher precision than int .

===============================================================

**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);

}

Error:

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

int result = a \*\* b;

^

Main.java:6: error: reached end of file while parsing

System.out.println(result);

Java is not allowed to use \*\*for exponential.

It use Math.pow(base,exponent);

Corrected Code:

import java.math.\*;

public class Main {

public static void main(String[] args) {

int a = 10;

int b = 5;

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

System.out.println(result);

}

}

^

Output:

100000

===============================================================

***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);

}

}

 **What is the output of this code? How does operator precedence affect the result?**

**Output:** 20

Multiplication has more precision than addition that’s why first multiplication is done. If there is no precedence then ans for the above task will be 30.

===================================================================

***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);

}

}

 **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

at Main.main(Main.java:5)

=================================================================

***Snippet 20:***

public class Main {

public static void main(String[] args) {

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

}

}

 **What syntax error occurs? How does the missing semicolon affect compilation?**

**error:** ';' expected

Semicolon is required to terminate the statements.

Corrected Code:

public class Main {

public static void main(String[] args) {

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

}

}

**Output:**

Hello, World

===================================================================

***Snippet 21:***

public class Main {

public static void main(String[] args) {

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

// Missing closing brace here

}

 **What does the compiler say about mismatched brace**

error: reached end of file while parsing

Here ‘ }’ is missing

=============================================================

***Snippet 22:***

public class Main {

public static void main(String[] args) {

static void displayMessage() {

System.out.println("Message");

}

}

}

 **What syntax error occurs? Can a method be declared inside another method?**

error: illegal start of expression

static void displayMessage() {

^

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

}

***================================================================***

***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");

}

}

}

 **Error to Investigate:** Why does the default case print after "Value is 2"? How can you prevent

the program from executing the default case?

**Output:**

Value is 2

Value is 3

Default case

We can prevent printing default case by adding ‘break’.

**Corrected Code:**

public class Confusion {

public static void main(String[] args) {

int value = 2;

switch(value) {

case 1:

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

break;

case 2:

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

break;

case 3:

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

break;

default:

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

}

}

}

**Output:**

Value is 2

==========================================================

***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");

}

}

}

 **Error to Investigate:** 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?

Because in case we are not breaking the statement so it will run all cases .By adding break it will go outside the loop

public class MissingBreakCase {

public static void main(String[] args) {

int level = 1;

switch(level) {

case 1:

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

break;

case 2:

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

break;

case 3:

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

break;

default:

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

}

}

}

D:\Feb 25\Day\_2Assignment>javac MissingBreakCase.java

D:\Feb 25\Day\_2Assignment>java MissingBreakCase

Level 1

===================================================================

***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:** incompatible types: possible lossy conversion from double to int

switch(score) {

Switch does not support double or float value. By doing a type casting it will work. But decimal places will be removed. So by using if else we can run program.

**Corrected Code:**

public class Switch {

public static void main(String[] args) {

double score = 85.0;

if(score==100) {

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

}

else if(score==85){

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

}

else{

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

}

}

}

**Output:**

**Great job!**

============================================================

***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: duplicate case label

Because switch doesn’t allow the same lable case. By merging two cases we can avoid error.

public class Switch {

public static void main(String[] args) {

int number = 5;

switch(number) {

case 5:

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

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

break;

default:

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

}

}

}

Output:

Number is 5

This is another case 5

=====================================================================

**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"

**Code:**

import java.util.\*;

public class Grade {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int score=sc.nextInt();

if(score>=90) {

System.out.println("A");

}

else if(score<90 && score>=80){

System.out.println("B");

}

else if(score<80 && score>=70){

System.out.println("C");

}

else if(score<70 && score>=60){

System.out.println("D");

}

else {

System.out.println("D");

}

}

}

**Output:**

D:\Feb 25>javac Grade.java

D:\Feb 25>java Grade

80

B

=====================================================================

**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.

**Ans:**

import java.util.\*;

public class DaysOfWeek{

public static void main(String args[]){

Scanner sc= new Scanner(System.in);

int day = sc.nextInt();

switch(day){

case 1:

System.out.println("Monday");

switch (1) { // Inner switch to determine weekday/weekend

case 1:

System.out.println("Weekday");

break;

}

break;

case 2:

System.out.println("Tuesday");

switch (1) { // Inner switch to determine weekday/weekend

case 1:

System.out.println("Weekday");

break;

}

break;

case 3:

System.out.println("Wednesday");

switch (1) { // Inner switch to determine weekday/weekend

case 1:

System.out.println("Weekday");

break;

}

break;

case 4:

System.out.println("Thursday");

switch (1) { // Inner switch to determine weekday/weekend

case 1:

System.out.println("Weekday");

break;

}

break;

case 5:

System.out.println("Friday");

switch (1) { // Inner switch to determine weekday/weekend

case 1:

System.out.println("Weekday");

break;

}

break;

case 6:

System.out.println("Saturday");

switch (1) { // Inner switch to determine weekday/weekend

case 1:

System.out.println("Weekday");

break;

}

break;

case 7:

System.out.println("Sunday");

switch (1) { // Inner switch to determine weekday/weekend

case 1:

System.out.println("Weekday");

break;

}

break;

}

}

}

Output:

D:\Feb 25>javac DaysOfWeek.java

D:\Feb 25>java DaysOfWeek

5

Friday

Weekday

====================================================================

**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 if

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

**Ans:**

import java.util.\*;

public class Calculator {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

System.out.println("Enter First Number");

double num1 =sc.nextDouble();

System.out.println("Enter an operator: +,-,\*,/");

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

System.out.println("Enter Second Number");

double num2 =sc.nextDouble();

double result;

switch(operator){

case '+':

result=num1 + num2;

System.out.println(num1+"+"+num2+"="+result);

break;

case '-':

result=num1 - num2;

System.out.println(num1+"-"+num2+"="+result);

break;

case '\*':

result=num1 + num2;

System.out.println(num1+"\*"+num2+"="+result);

break;

case '/':

if(num2==0){

System.out.println("error:Division by zero is not allowed");

}

else {

result = num1 / num2;

System.out.println(num1+"/"+num2+"="+result);

}

break;

default:

System.out.println("Invalid input...Please enter the valid input");

}

}

}

Output:

D:\Feb 25>javac Calculator.java

D:\Feb 25>java Calculator

Enter First Number

20

Enter an operator: +,-,\*,/

+

Enter Second Number

30

20.0+30.0=50.0

**=================================================================**

**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%.

Program-

import java.util.\*;

public class Discount{

public static void main(String[] args){

Scanner sc = new Scanner (System.in);

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

double totalPurchase=sc.nextDouble();

double finalAmount=0;

System.out.println("Do you have a membership?(if yes enter 1,else enter 0)");

int membership=sc.nextInt();

if(totalPurchase>=1000){

finalAmount=totalPurchase-(totalPurchase\*(20.0/100));

if(membership==1){

finalAmount=finalAmount-(totalPurchase\*(5.0/100));

System.out.println(finalAmount);

}

else

System.out.println(totalPurchase);

}

else if(totalPurchase>=500 && totalPurchase<1000){

finalAmount=totalPurchase-(totalPurchase\*(10.0/100));

if(membership==1){

finalAmount=finalAmount-(totalPurchase\*(5.0/100));

System.out.println(finalAmount);

}

else

System.out.println(totalPurchase);

}

else{

finalAmount=totalPurchase-(totalPurchase\*(5.0/100));

if(membership==1){

finalAmount=finalAmount-(totalPurchase\*(5.0/100));

}

else

System.out.println(finalAmount);

}

}

}

Output:

D:\Feb 25\Day\_2Assignment>java Discount

Enter your total purchase amount:

1200

Do you have a membership?(if yes enter 1,else enter 0)

1

900.0

**=====================================================================**

**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.

**Ans:**

import java.util.\*;

public class Result{

public static void main(String args[]){

Scanner sc= new Scanner(System.in);

System.out.println("Enter the marks of 1st subject: ");

int sub1=sc.nextInt();

System.out.println("Enter the marks of 2nd subject: ");

int sub2=sc.nextInt();

System.out.println("Enter the marks of 3rd subject: ");

int sub3=sc.nextInt();

int failcount=0;

failcount += (sub1 < 40) ? 1 : 0;

failcount += (sub2 < 40) ? 1 : 0;

failcount += (sub3 < 40) ? 1 : 0;

switch( failcount){

case 0:

System.out.println("congratulations!!! you are passed");

break;

case 1:

System.out.println("You are failed in 1 Subject");

break;

case 2:

System.out.println("You are failed in 2 Subject");

break;

default:

System.out.println("You are failed in 3 Subject");

}

}

}

**Output:**

D:\Feb 25>javac Result.java

D:\Feb 25>

D:\Feb 25>java Result

Enter the marks of 1st subject:

50

Enter the marks of 2nd subject:

80

Enter the marks of 3rd subject:

60

congratulations!!! you are passed

=================================================================