Lab Assignment 2

**Snippet**  1:

public class Main {

public void main(String[] args) {

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

}

}

 What error do you get when running this code?

Runtime error:Main method is not static in class Main,

**Correct**  **Snippet** :

public class Main

{

public static void main(String args[])

{

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

}

}

**Snippet**  2:

public class Main {

static void main(String[] args) {

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

}

}

 What happens when you compile and run this code?

Runtime error:as Main method is not declare as and not found in

Class

**Correct**  **Snippet** :

public class Main

{

public static void main(String args[])

{

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

}

}

**Snippet**  3:

public class Main {

public static int main(String[] args) {

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

return 0;

}

}

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

Error: Main method must return a value of type void in class Main,

The keyword void tells Java that the main method won't return a value

**Correct**  **Snippet** :

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!");

}

}

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

Error: Main method not found in class Main

String[] args is needed to pass a command line arguments in java

**Correct**  **Snippet** :

public class Main

{

public static void main(String args[])

{

System.out.println("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");

}

}

 Can you have multiple main methods? What do you observe?

Yes we can have multiple main methods but there should be have only one main method with String[] args

**correct**  code :

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");

}

}

------------------------------------------------------------------------------------------------------

**Snippet**  6:

public class Main {

public static void main(String[] args) {

int x = y + 10;

System.out.println(x);

}

}

 What error occurs? Why must variables be declared?

Error: Main.java:5: error: cannot find symbol

int x = y + 10;

^

symbol: variable y

location: class Main

1 error

Variables are declare to store the data in memory loction

**Correct**  **Snippet** :

public class Main

{

public static void main(String[] args)

{

int y=10;

int x = y + 10;

System.out.println(x);

}

}

-----------------------------------------------------------------------------------------------------------

**Snippet**  7:

public class Main {

public static void main(String[] args) {

int x = "Hello";

System.out.println(x);

}

}

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

Error:

Main.java:3: error: incompatible types: String cannot be converted to int

int x = "Hello";

^

1 error

Type safety checks that the object a method may try tomanipulate is of **correct**  type

**correct**  **Snippet** :

public class Main

{

public static void main(String[] args)

{

String x = "Hello";

System.out.println(x);

}

}

----------------------------------------------------------------------------------------------------------------

**Snippet**  8:

public class Main {

public static void main(String[] args) {

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

}

}

 What syntax errors are present? How do they affect compilation?

Error: closing parenthesis and semi colon are missing from the line("Hello, World!"

Syntax error prevent a program to compile and execute properly, A program will not compile until all syntax errors are **correct** ed.

**correct**  **Snippet** :

public class Main

{

public static void main(String[] args)

{

System.out.println("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?

Error occured due to reserved keywords(Class) used to declare a variable.

Reserved keywords are reserved for special purpose.

Error:

Main.java:5: error: not a statement

int class = 10;

^

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

int class = 10;

^

Main.java:5: error: <identifier> expected

int class = 10;

^

Main.java:6: error: illegal start of expression

System.out.println(class);

^

Main.java:6: error: <identifier> expected

System.out.println(class);

^

5 errors

**correct**  **Snippet** :

public class Main

{

public static void main(String[] args)

{

int a = 10;

System.out.println(a);

}

}

------------------------------------------------------------------------------------------------------------------------

**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?

we have to create an object of the class to refer to a non-static variable from a static context.

Error:

Main.java:9: error: 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);

^

2 errors

----------------------------------------------------------------------------------------------------------------------------

**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?

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

at Main.main(Main.java:4)

error occured because ther is no elementat the 5th index of array.

**correct** :

public class Main {

public static void main(String[] args) {

int[] arr = {1, 2, 3,0,4,7};

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

}

}

--------------------------------------------------------------------------------------------------------------------------------

**Snippet**  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?

The while loop executes infinite loops

to avoid infinite loops break statement is used

**correct**  **Snippet** :

public class Main

{

public static void main(String[] args)

{

while (true)

{

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

break;

}

}

}

---------------------------------------------------------------------------------------------------------------------------

**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?

NullPointerException is thrown. It occurs due to there is no valueassign to str.

error: Cannot invoke "String.length()" because "<local1>" is null

at Main.main(Main.java:4)

**correct**  **Snippet** :

public class Main

{

public static void main(String[] args)

{

String str = null;

if (str != null)

{

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

}

else

{

System.out.println("String is null");

}

}

}

-------------------------------------------------------------------------------------------------------------------------------

**Snippet**  14:

public class Main {

public static void main(String[] args) {

double num = "Hello";

System.out.println(num);

}

}

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

Error: error: incompatible types: String cannot be converted to double

double num = "Hello";

**correct**  **Snippet** :

public class Main {

public static void main(String[] args) {

String num = "Hello";

System.out.println(num);

}

}

------------------------------------------------------------------------------------------------------------------------------

**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);

}

}

 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

int result = num1 + num2;

**correct**  **Snippet** :

public class Main {

public static void main(String[] args) {

int num1 = 10;

double num2 = 5.5;

double result = num1 + num2;

System.out.println(result);

}

}

---------------------------------------------------------------------------------------------------------------------------------

**Snippet**  16:

public class Main {

public static void main(String[] args) {

int num = 10;

double result = num / 4;

System.out.println(result);

}

}

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

Output is :2.0 not expected.

**correct**  **Snippet**  is:

public class Main

{

public static void main(String[] args)

{

float num = 10f;

double result = num / 4;

System.out.println(result);

}

}

-----------------------------------------------------------------------------------------------------------------------------------------

**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);

}

}

 What compilation error occurs? Why is the \*\* operator not valid in Java?

Error: illegal start of expression

int result = a \*\* b;

^

1 error

**Correct**  **Snippet** :

public class Main {

public static void main(String[] args) {

int a = 10;

int b = 5;

int result = a \* b;

System.out.println(result);

}

}

------------------------------------------------------------------------------------------------------------------------------------------------

**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 is 20

-------------------------------------------------------------------------------------------------------------------------------------------------

**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)

public class Main {

public static void main(String[] args) {

int a = 10;

int b = 0;

try{

int result = a / b;

System.out.println(result);

}

catch (ArithmeticException e) {

System.out.println("Divided by zero operation cannot possible");

}

}

}

----------------------------------------------------------------------------------------------------------------------------------------------------

**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:Main.java:3: error: ';' expected

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

^

1 error

**correct**  **Snippet** :

public class Main {

public static void main(String[] args) {

System.out.println("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 braces?

Error: reached end of file while parsing

}

^

1 error

**correct**  **Snippet** :

public class Main {

public static void main(String[] args) {

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

}

// Missing closing brace here

}

----------------------------------------------------------------------------------------------------------------------------------------------------

**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, enum, or record expected

}

^

2 errors

**correct**  **Snippet** :

public class Main {

static void displayMessage() {

System.out.println("Message");

}

public static void main(String[] args){

displayMessage();

}

}

-------------------------------------------------------------------------------------------------------------------------------------------------

**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?

error: class Confusion is public, should be declared in a file named Confusion.java

public class Confusion {

^

1 error

**correct**  **Snippet** :

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");

}

}

}

-------------------------------------------------------------------------------------------------------------------------------

**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?

**correct**  **Snippet** :

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");

}

}

}

---------------------------------------------------------------------------------------------------------------------------------------

**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?

error: patterns in switch statements are a preview feature and are disabled by default.

switch(score) {

^

(use --enable-preview to enable patterns in switch statements)

Switch.java:5: error: constant label of type int is not compatible with switch selector type double

case 100:

^

Switch.java:8: error: constant label of type int is not compatible with switch selector type double

case 85:

^

3 errors

error: compilation failed

**correct**  **Snippet** :

public class Switch {

public static void main(String[] args) {

int score = 85;

switch(score) {

case 100:

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

break;

case 85:

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

break;

default:

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

}

}

}

---------------------------------------------------------------------------------------------------------------------------

**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?

Case 5 is duplicate it must be a constant

Switch.java:8: error: duplicate case label

case 5:

^

1 error

error: compilation failed

**Correct**  **Snippet** :

public class Switch {

public static void main(String[] args) {

int number = 5;

switch(number) {

case 5:

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

break;

case 9:

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

break;

default:

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

}

}

}

-------------------------------------------------------------------------------------------------------------------------------------------

**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:

public class Studentgrade{

public static void main(String[] args) {

int score = 82;

switch(score/10) {

case 10:

case 9:

System.out.println("Grade = A");

break;

case 8:

System.out.println("Grade = B");

break;

case 7:

System.out.println("Grade = C");

break;

case 6:

System.out.println("Grade = D");

break;

default:

System.out.println("Grade = F");

break;

}

}

}

---------------------------------------------------------------------------------------------------------------------------

**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.**

Public class Days

{

public static void main(String args[])

{

int day=1;

switch(day)

{

case 1:

System.out.println("Today is monday");

System.out.println("weekday");

break;

case 2:

System.out.println("Today is Tuesday");

System.out.println("weekday");

break;

case 3:

System.out.println("Today is wednesday");

System.out.println("weekday");

break;

case 4:

System.out.println("Today is Thursday");

System.out.println("weekday");

break;

case 5:

System.out.println("Today is Friday");

System.out.println("weekend");

break;

case 6:

System.out.println("Today is Saturday");

System.out.println("weekend");

break;

case 7:

System.out.println("Today is Sunday");

System.out.println("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) {

char ope;

Double number1, number2, result;

Scanner input = new Scanner(System.in);

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

ope = input.next().charAt(0);

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

number1 = input.nextDouble();

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

number2 = input.nextDouble();

switch (ope) {

case '+':

result = number1 + number2;

System.out.println(number1 + " + " + number2 + " = " + result);

break;

case '-':

result = number1 - number2;

System.out.println(number1 + " - " + number2 + " = " + result);

break;

case '\*':

result = number1 \* number2;

System.out.println(number1 + " \* " + number2 + " = " + result);

break;

case '/':

result = number1 / number2;

System.out.println(number1 + " / " + number2 + " = " + result);

break;

default:

System.out.println("Invalid operator!");

break;

}

input.close();

}

}

-----------------------------------------------------------------------------------------------------------------------------------------

**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.**

Source code:

class Discount {

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

System.out.println("Enter price ");

int price =sc.nextInt();

System.out.println("Enter 1 if membership present else 0");

int membership = sc.nextInt();

if(price >=1000){

if(membership==1){

System.out.println("Discount is : " + ((price\*25)/100));

}else{

System.out.println("Discount is : "+(price\*20)/100);

}

}else if(price >=500 && price<=999){

if(membership==1){

System.out.println("Discount is : "+(price\*15)/100);

}else{

System.out.println("Discount is : "+(price\*10)/100);

}

}else{

if(membership==1){

System.out.println("Discount is : "+(price\*10)/100);

}else{

System.out.println("Discount is : "+(price\*5)/100);

}

}

}

}

-------------------------------------------------------------------------------------------------------------------------------------------

**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 StudentGrades {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter grades for three subjects:");

System.out.print("Subject 1: ");

int grade1 = sc.nextInt();

System.out.print("Subject 2: ");

int grade2 = sc.nextInt();

System.out.print("Subject 3: ");

int grade3 = sc.nextInt();

int failedSubjects = 0;

if (grade1 < 40) {

failedSubjects++;

}

if (grade2 < 40) {

failedSubjects++;

}

if (grade3 < 40) {

failedSubjects++;

}

if (failedSubjects == 0) {

System.out.println("Student Passed!");

} else {

System.out.println("Student Failed in " + failedSubjects + " subjects.");

}

}

}