CDAC Mumbai

Lab Assignment

Section 1: Error-Driven Learning in Java

Objective: This assignment focuses on understanding and fixing common errors encountered in Java programming. By analyzing and correcting the provided code snippets, you will develop a deeper understanding of Java's syntax, data types, and control structures.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Snippet 1:

public class Main{

public void main(String args[])

{

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

}

}

What error do you get when running this code?

ANS:=>The compiler shows main method should be static,because we have to no need to create object of main method, so we can call the main method witjout creating its object.

Snippet 2:

public class Main {

static void main(String[] args) {

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

}

}

What happens when you compile and run this code?

ANS;=> Compiler shows the error that main method is not found, we need to define the main method in predefined way.

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?

ANS:=>The compiler shows the error that main method should be void.Because main method doesn’t return anything.

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?

ANS:=> it gives compile time error,because main method accept the input as an String array of arguments.

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?

ANS:=>Yes , we can have multiple main methods in java with different parameter list but jvm can call only that method which have predefined main method signature.

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?

ANS;=> cant find symbol y ,the compiler shows the error. Variable must be declared because it allocates the memory during compilation. So we can store the value for that particular variable.

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?

Ans;=> Incompatible types , String can not be converted to int error gives.

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?

ANS:=>reached end of file while parsing, and ‘ ) ’ expected. We need to close the sop statement. So the compiler get an idea where statement ended.

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?

ANS:=>

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:3: error: illegal start of type

int class = 10;

^

Main.java:3: error: <identifier> expected

int class = 10;

^

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

int class = 10;

^

Main.java:4: error: illegal start of type

System.out.println(class);

^

Main.java:4: error: ';' expected

System.out.println(class);

^

Main.java:4: error: invalid method declaration; return type required

System.out.println(class);

^

Main.java:4: error: illegal start of type

System.out.println(class);

^

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

}

^

11 errors

We cant use reserved keywords as an identifiers in java program because they have predefined meaning within language syntax and reserved for specific purpose.

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?

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);

method overloading is not allowed because here we not declared the method as static so we cant access it without object.

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?

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5

Exception is occur because we are accessing the element at index 5 but there is only element present upto index 2.

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?

ANS:=> in while loop the condition is true ,so it will compile continuous infinitely ,there is no break statement. We need to give some condition.

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?

Exception in thread "main" java.lang.NullPointerException because we are calculating the length of null string.

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?

incompatible types: String cannot be converted to double

double num = "Hello";

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?

incompatible types: possible lossy conversion from double to int

int result = num1 + num2; we handle different data types in operation by doing type casting .so we can handle the losses.

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?

2.0 yes

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?

illegal start of expression

int result = a \*\* b;

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?

20 , expression can be solved as per the operator precedence.

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?

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

Because any no divided by zero causes infinity that’s why it causes issue.

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

System.out.println("Hello, World") semicolon tells the compiler that statement ended.

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?

reached end of file while parsing

}

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?

illegal start of expression

static void displayMessage() {

^

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

static void displayMessage() {

^

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

static void displayMessage() {

yes we can declared a method inside another method by using local classes or lambda expression.

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?

Default case is print after case 2 because we have given in switch statement int variable value is 2. We can prevent the program from executing the default case by using break statement after case 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?

ANS;=> the value of int variable level is 1, so it start from case 1 and there is no break statement after case so it continuously goes upto default statement.

Break statement terminate the execution after some perfoeming operation.

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;=> incompatible types: possible lossy conversion from double to int

switch(score) { switch statement can accept only int value ,here the value of score is double data type, so automatic type casting from large data type to small data type is not allowed. We can change the data type to int of score so it will work.

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;=>Because compiler don’t accept duplicate values, so it will generate ambiguity that’s why we cant use two identical case labels.

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"

import java.util.Scanner;

public class Grades

{

public static void main(String args[])

{

int score;

Scanner sc=new Scanner(System.in);

System.out.println("Enter Student score:");

score=sc.nextInt();

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 if(score<=60)

{

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.

import java.util.Scanner;

public class DaysOfWeek

{

public static void main(String args[])

{

int day;

Scanner sc=new Scanner(System.in);

System.out.println("Enter the date");

day=sc.nextInt();

switch(day)

{

case 1:

System.out.println("Monday");

System.out.println("Weekday");

break;

case 2:

System.out.println("Tuesday");

System.out.println("Weekday");

break;

case 3:

System.out.println("Wednesday");

System.out.println("Weekday");

break;

case 4:

System.out.println("Thursday");

System.out.println("Weekday");

break;

case 5:

System.out.println("Friday");

System.out.println("Weekday");

break;

case 6:

System.out.println("Saturday");

System.out.println("Weekday");

break;

case 7:

System.out.println("Sunday");

System.out.println("Weekend");

break;

default:

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

}

}

}

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 sc=new Scanner(System.in);

System.out.println("Enter First No");

int a=sc.nextInt();

System.out.println("Enter Second No:");

int b=sc.nextInt();

System.out.println("Enter the Operator");

char op=sc.next(".").charAt(0);

int res=0;

switch(op)

{

case '+':

res=a+b;

System.out.println(res);

break;

case '-':

res=a-b;

System.out.println(res);

break;

case '\*':

res=a\*b;

System.out.println(res);

break;

case '/':

if(b==0)

{

System.out.println("A no can not be devided by zero");

}else

{

res=a/b;

System.out.println(res);

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%.

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.