

Faculty of Computer Applications and Information Technology
M. Sc. (IT) – Sem 2
Practicals of Mobile Application Development using Flutter
Assignment 1

Note: Kindly submit your assignment on or before 21st January 2026 on Moodle.

Kindly prepare a PDF of the assignment solution. Save the PDF using your Enrollment Number as the file name and upload it to Moodle.

1. Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included).
2. Write a program to check if a number is a prime number.
3. Write a program that accepts a sentence and calculate the number of letters and digits. Suppose the following input is supplied to the program: hello world! 123 Then, the output should be: LETTERS 10 DIGITS 3
4. Write a program to calculate squares of even numbers between a given range (like 1 to 30).
5. Write a Dart program to accept age input from the user. If the entered age is less than 18, throw a custom exception. Properly handle both FormatException (invalid input) and the custom exception using try-catch.
6. Create a Dart program to implement a To-Do List application using a List.
The program should support the following operations:
 - Add new tasks
 - Remove existing tasks
 - Mark tasks as completed
 - Display pending tasks and completed tasks separately
7. Develop a Dart program to simulate basic bank account operations.
Create a BankAccount class that uses encapsulation to protect account balance.
Implement the following methods:
 - deposit() to add amount to the balance

- withdraw() to deduct amount (handle insufficient balance condition)
 - checkBalance() to display the current balance
 - Ensure the balance variable is declared as private.
8. Create a Dart program to calculate salaries for different types of employees.
- Define a base class Employee and derive two subclasses:
PermanentEmployee
ContractEmployee
 - Override the calculateSalary() method in each subclass to compute salary according to the employee type.

=====

1. Given Code

```
class Student {  
    int id;  
    String name;  
  
    Student(int id, String name) {  
        id = id;  
        name = name;  
    }  
}  
  
void main() {  
    Student s = Student(1, "Rahul");  
    print(s.name);  
}
```

Tasks -

- Does the code produce output or error?
- If error → identify and rectify it
- Add functionality to:
 - Display both id and name
 - Add a method display()

2. Given Code

```
void main() {  
    int marks = 85;  
  
    if (marks > 90) {  
        print("A");  
    }
```

```
    } else if (marks > 75) {  
        print("B");  
    } else {  
        print("C");  
    }  
}
```

Tasks-

- Predict output
- Identify logical issue
- Fix grading logic
- Add functionality: Accept marks from user.

3. Given Code -

```
class Product {  
    String name;  
    double price;  
  
    Product(this.name, this.price);  
}  
  
void main() {  
    Product p = Product("Laptop", -50000);  
    print(p.price);  
}
```

Tasks -

- Is this logically correct?
- Add validation using exception handling
- Add functionality: Apply 10% discount if price > 30000

4. Given Code -

```
class Person {  
    String name;  
    Person(this.name);  
}  
  
void main() {  
    Person p1 = Person("Amit");  
    Person p2 = Person("Amit");  
  
    print(p1 == p2);  
}
```

Tasks-

- Predict output
- Explain why
- Override equality operator
- Add functionality: compare by name

5. Given Code -

```
void main() {  
    int day = 1;  
  
    switch (day) {  
        case 1:  
            print("Monday");  
        case 2:  
            print("Tuesday");  
            break;  
    }  
}
```

Tasks -

- Predict output
- Explain Dart switch behavior
- Fix the logic
- Add functionality: handle all weekdays