



Bachelor of Information Technology (Hons) Assignment Cover Sheet

Course Code: EC3243 Course Title: Object Oriented Software Development
Assignment Title: Practical Report Due Date: 04/November/2025
Date Submitted: 04/November/2025 Lecturer Name: Mr. Subhash Paudel

To be completed if this is an individual assignment

I declare that this assignment is my individual work. I have not worked collaboratively nor have I copied from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for me by another person.

Student Name: Suraj Khadka

Student ID: 00021525

Signature: Suraj

To be completed if this is a group assignment

We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for us by another person.

Student ID

Student Name

Signature

Lecturer's comments:

Total Marks: _____

Lecturer's Signature: _____

Feedback to Student:

I/We acknowledged receiving feedback from the lecturer on this assignment.

Student's Signature: _____

Extension certification:

This assignment has been given an extension and is now due on _____.

Lecturer's Signature: _____

Table of Contents:

Source code of Bank Account:	10
Incorrect Outcome:	12
Marking Scheme	13

Source Code:

```
#include <iostream>
#include <string>
using namespace std;

// =====
// Base Class: Cricketer
// =====

class Cricketer{
protected:
    string name;
    int age;
    int numMatches;

public:
    // Constructor
    Cricketer(string n, int a, int m) {
        name = n;
        age = a;
        numMatches = m;
        cout << "Cricketer " << name << " created.\n";
    }
}
```

```
// Virtual function for displaying player details
virtual void display() {
    cout << "-----\n";
    cout << "Player Name : " << name << endl;
    cout << "Age : " << age << endl;
    cout << "Matches Played : " << numMatches << endl;
}

// Virtual destructor
virtual ~Cricketer() {
    cout << "Cricketer " << name << " destroyed.\n";
}
};

// =====
// Derived Class: Batsman
// =====
class Batsman : public Cricketer {
private:
    int runs;
    int halfCenturies;
    int centuries;

public:
    // Constructor
    Batsman(string n, int a, int m, int r, int half, int cen)
        : Cricketer(n, a, m) {
```

```
    runs = r;
    halfCenturies = half;
    centuries = cen;
}

// Override display()
void display() override {
    Cricketer::display();
    cout << "Total Runs    :" << runs << endl;
    cout << "Half Centuries  :" << halfCenturies << endl;
    cout << "Centuries     :" << centuries << endl;
    cout << "-----\n";
}

// Destructor
~Batsman() {
    cout << "Batsman " << name << " destroyed.\n";
}

// =====
// Derived Class: Bowler
// =====
class Bowler : public Cricketer {
private:
    int wickets;
    int fiveWickets;
    int hatTricks;
```

```
public:  
    // Constructor  
  
    Bowler(string n, int a, int m, int w, int five, int hat)  
        : Cricketer(n, a, m) {  
  
        wickets = w;  
        fiveWickets = five;  
        hatTricks = hat;  
    }  
  
    // Override display()  
  
    void display() override {  
  
        Cricketer::display();  
  
        cout << "Total Wickets : " << wickets << endl;  
        cout << "Five Wicket Hauls: " << fiveWickets << endl;  
        cout << "Hat-Tricks : " << hatTricks << endl;  
        cout << "-----\n";  
    }  
  
    // Destructor  
  
    ~Bowler() {  
        cout << "Bowler " << name << " destroyed.\n";  
    }  
};  
  
// ======  
// Main Function  
// ======
```

```

int main() {

    cout << "=====\\n";
    cout << " AUSTRALIAN CRICKET TEAM DATA \\n";
    cout << "=====\\n\\n";

    // Creating multiple objects for both derived classes
    Batsman batsman1("Steve Smith", 35, 190, 9500, 45, 32);
    Batsman batsman2("David Warner", 38, 170, 8800, 41, 27);

    Bowler bowler1("Mitchell Starc", 33, 120, 310, 8, 2);
    Bowler bowler2("Pat Cummins", 32, 100, 280, 6, 1);

    cout << "\\n===== BATSMEN DETAILS =====\\n";
    batsman1.display();
    batsman2.display();

    cout << "\\n===== BOWLERS DETAILS =====\\n";
    bowler1.display();
    bowler2.display();

    cout << "\\nEnd of program. Destroying all objects...\\n";
    return 0;
}

```

Program Outcome:

BATS MEN Details:

===== BATSMEN DETAILS =====

Player Name : Steve Smith
Age : 35
Matches Played : 190
Total Runs : 9500
Half Centuries : 45
Centuries : 32

Player Name : David Warner
Age : 38
Matches Played : 170
Total Runs : 8800
Half Centuries : 41
Centuries : 27

===== BOWLERS DETAILS =====

Player Name : Mitchell Starc
Age : 33
Matches Played : 120
Total Wickets : 310
Five Wicket Hauls: 8
Hat-Tricks : 2

Player Name : Pat Cummins

Bowler Details:

===== BOWLERS DETAILS =====

Player Name : Mitchell Starc
Age : 33
Matches Played : 120
Total Wickets : 310
Five Wicket Hauls: 8
Hat-Tricks : 2

Player Name : Pat Cummins
Age : 32
Matches Played : 100
Total Wickets : 280
Five Wicket Hauls: 6
Hat-Tricks : 1

End of program. Destroying all objects...
Bowler Pat Cummins destroyed.
Cricketer Pat Cummins destroyed.
Bowler Mitchell Starc destroyed.
Cricketer Mitchell Starc destroyed.
Batsman David Warner destroyed.
Cricketer David Warner destroyed.
Batsman Steve Smith destroyed.
Cricketer Steve Smith destroyed.

Source code of Bank Account:

```
#include <iostream>
#include <stdexcept>
using namespace std;

class BankAccount{
private:
    double balance;

public:
    // Constructor
    BankAccount(double bal) {
        balance = bal;
    }

    // Method to read withdrawal amount
    void readWithdrawAmount() {
        double withdraw;
        cout << "Enter withdrawal amount: ";
        cin >> withdraw;

        try {
            if (withdraw <= balance) {
                balance -= withdraw;
                cout << "Withdrawal successful.\n";
                cout << "Remaining Balance: " << balance << endl;
            } else {

```

```
        throw runtime_error("Error: Withdrawal amount exceeds balance!");

    }

} catch (runtime_error &e) {

    cout << e.what() << endl;

}

}

void readBalance() {

    cout << "Current Balance: " << balance << endl;

}

// Main Function

int main() {

    cout << "==== Bank Account Program ===\n\n";

    double initialBalance;

    cout << "Enter initial balance: ";

    cin >> initialBalance;

    BankAccount acc(initialBalance);

    acc.readBalance();

    acc.readWithdrawAmount();

    return 0;

}
```

Correct Outcome:

```
==== Bank Account Program ===

Enter initial balance: 10000
Current Balance: 10000
Enter withdrawal amount: 5000
Withdrawal successful.
Remaining Balance: 5000

-----
Process exited after 19.1 seconds with return value 0
Press any key to continue . . .
```

Incorrect Outcome:

```
C:\Users\DELL\OneDrive\Desktop > + | v
==== Bank Account Program ===

Enter initial balance: 50000
Current Balance: 50000
Enter withdrawal amount: 60000
Error: Withdrawal amount exceeds balance!

-----
Process exited after 9.35 seconds with return value 0
Press any key to continue . . .
```

Marking Scheme

Name:- Suraj Khadka

Student id:- 00021525

Criteria	Marks Allocated	Marks Obtained	Comments
Usage of Proper Class	10 (5 marks each)		
Implementation of Objects in the Main Program	10 (5 marks each)		
Output - Correct Outcome - Screenshot and Code -Output with correct data	20 (10 marks each)		
Demonstration and Viva	10 (5 marks each)		