

```

using System;
namespace Week4
{
    class Program
    {
        static void Main()
        {
            /*
            Input validation for the Banking Details.
            -Make sure all the input can't be blank.
            -Prompt user to enter the Bank Account ID, Amount to deposit,
            Term (in Months) and the Post Code.
            -The account ID should be seven characters long.
            -The first three characters of ID should NOT be either "AAA" or "AAB" or any upper or lower case
            arrangement of these.
            -The last four characters should all be digits (a space is not a digit).
            -The amount should be convertible to a Decimal.
            -It should be between $0.00 and $10,000.00 inclusive.
            -The term should be an integer (without rounding).
            -The term should be either: 1, 2, 3, 4, or 5 or a non-zero multiple of 6.
            -The postcode must consist of 4 digits (a space is not a digit).
            -The first digit must not be 1, 8 or 9.
            -The valid post code the following state should be show in the confirmation output
            First Digit State
            0 NT
            2 NSW
            3 VIC
            4 QLD
            5 SA
            6 WA
            7 TAS
            Data Type Validation */

            string id = "";
            Console.Write("Enter Account ID: ");
            id = Console.ReadLine();
            if (id.Length != 7 || !Char.IsLetter(id[0]) || !Char.IsLetter(id[1]) || !Char.IsLetter(id[2])
                || id.ToUpper().StartsWith("AAA") || id.ToUpper().StartsWith("AAB")
                || !Char.IsDigit(id[3]) || !Char.IsDigit(id[4]) || !Char.IsDigit(id[5])
                || !Char.IsDigit(id[6]))
            {
                Console.WriteLine("Invalid ID... It must be 7 characters\nFirst three characters can't be AAA
                or AAB upper or lower\nLast 4 characters must be digits\n\n");
                return;
            }

            decimal amount = 0;
            Console.Write("Enter Deposit Amount: $ ");
            if (decimal.TryParse(Console.ReadLine(), out amount) == false || amount < 0 || amount > 10000)
            {
                Console.WriteLine("Amount should be 0-10000 only - try again");
                return; ;
            }

            int term = 0;
            Console.Write("Enter Deposit Term (1,2,3,4,5 or 6x) month(s)? ");
            if (int.TryParse(Console.ReadLine(), out term) == false || term < 1
                || (term > 6 && term % 6 != 0))
            {
                Console.WriteLine("Invalid term - try again");
                return;
            }

            string postcode = "", state = "";
            Console.Write("Enter post code: ");
            postcode = Console.ReadLine();
            if (postcode.Length != 4 || !Char.IsDigit(postcode[0]) || !Char.IsDigit(postcode[1])
                || !Char.IsDigit(postcode[2]) || !Char.IsDigit(postcode[3]))
            {
                Console.WriteLine("Invalid post code - try again");
                return;
            }
        }
    }
}

```

```
switch (postcode[0])
{
    case '0': state = "NT"; break;
    case '2': state = "NSW"; break;
    case '3': state = "VIC"; break;
    case '4': state = "QLD"; break;
    case '5': state = "SA"; break;
    case '6': state = "WA"; break;
    case '7': state = "TAS"; break;
    default: Console.WriteLine("Invalid state - try again"); break;
}

// output
Console.WriteLine("{0} Deposit {1:c} for {2} month(s) at Branch Post Code: {3} - {4}\n\n",
    id, amount, term, postcode, state);
}
}
```