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

const int DefaultNumberOfBalls = 4;

enum IONotificationEnum {
    ObjectNametNotification =1,
    ObjectSizeNotification =2,
    ObjectName =3,
    ObjectSize = 4
};

string GetObjectNotificationString(IONotificationEnum notify) {

    switch (notify) {
    case IONotificationEnum::ObjectNametNotification:
            return "Enter the ";
            break;
    case IONotificationEnum::ObjectSizeNotification:
        return "Enter size of Object ";
        break;
    case IONotificationEnum::ObjectName:
        return "Object Name: ";
        break;

    case IONotificationEnum::ObjectSize:
        return "Object Size: ";
        break;

    default:
        return "Invalid";
        break;

    }
}

string GetTheStyledAlert(string firstPart, string second , string third="  Object Name: ") {
    return firstPart + second + third;
}

IONotificationEnum GetObjectNameNotify() {
    return IONotificationEnum::ObjectName;
}

IONotificationEnum GetObjectSizeNotify() {
    return IONotificationEnum::ObjectSize;
}
```

```cpp

IONotificationEnum GetObjectSizeNotification() {
    return IONotificationEnum::ObjectSizeNotification;
}

IONotificationEnum GetObjectNametNotification() {
    return IONotificationEnum::ObjectNametNotification;
}



struct Ball {
    string Name;
    int size;
};

//Data Structure for Ball Objects

Ball ObjectBalls[4];


//_____________________Helper Functions ---------------------------------
//

string GetObjectPrompt(const string& notification, int number) {
    return notification + to_string(number);
}

// constexpr is a compile-time constant
template <typename T, size_t N>
constexpr size_t GetArraySize(T(&)[N]) {
    return N;
}
int GetNumberOfItemsInArray( string arr[], int size) {
    return size;
}

 string getOrdinalText(int number) {
    // Array of ordinal text representations
      string ordinals[] = {
        "First", "Second", "Third", "Fourth", "Fifth",
        "Sixth", "Seventh", "Eighth", "Ninth", "Tenth"
     };
     int _length = GetNumberOfItemsInArray(ordinals, GetArraySize
       (ordinals));

    // Check if the number is within the valid range
    if (number >= 1 && number <= _length) {
        return ordinals[number - 1];
```

```
 97      }
 98      else {
 99          return "Invalid";
100      }
101  }
102
103  string ReadAString( string message) {
104
105      string _object = "";
106      cout << message << endl;
107      getline(cin, _object);
108
109      return _object;
110  }
111
112
113  int ReadIntegerNumber( string message) {
114      cout << message << endl;
115      int _number;
116      cin >> _number;
117
118      return _number;
119  }
120
121
122  Ball ReadBallObject(int index) {
123      Ball object;
124
125      string s = GetTheStyledAlert(GetObjectNotificationString
           (GetObjectNametNotification()), getOrdinalText(index));
126      string propmptName = GetObjectPrompt(s, index);
127      string promptSize = GetObjectPrompt(GetObjectNotificationString
           (GetObjectSizeNotification()), index);
128      object.Name = ReadAString(propmptName);
129      object.size = ReadIntegerNumber(promptSize);
130      return object;
131  }
132
133  void ReadAllObjectBalls() {
134
135      int size = DefaultNumberOfBalls;
136      for (int i = 0; i < DefaultNumberOfBalls; i++) {
137          ObjectBalls[i] = ReadBallObject(i+1);
138          cin.ignore(); //Ignore the newline character left in the input
               buffer
139      }
140  }
141
142
```

```
143 void PrintAllObjectBallInfo() {
144
145     int size =  DefaultNumberOfBalls;
146     for (int i = 0; i < size; i++) {
147         cout << GetObjectNotificationString(GetObjectNameNotify()) <<
              ObjectBalls[i].Name << "\t";
148         cout << GetObjectNotificationString(GetObjectSizeNotify()) <<
              ObjectBalls[i].size << endl;
149     }
150 }
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168 /*
169 *
      ------------------------------------------------------------------------
      ------------------------------------
170 Problem Description:
171
172  In the project file, there is an image titled "ImageBall" that
       illustrates variants balls with their names and sizes.
173  Your task is to develop a program that reads the four balls with their
       names and sizes and get the the smallest ball size and the largest one,
174  plus the difference between the the largest and smallest
175 ------------------------------------------------------------------------
      ----------------------------------------
176 Requirements:
177  read the name of the balls with their sizes and them get the max , min ,
       difference beteen max and min of them
178
179 ------------------------------------------------------------------------
      -------------------------
180 Problem Analysis-:
181 Inputs:
182 - four objects of balls that contains name and size
```

```
183 -
184
185 Outputs:
186 - get max, min, difference beteen the max size and min size of the ball
187
188 Constraints:
189
190 Constant Values:
191
192
193
194 Decomposing problem:
195
196
197
198 Design Phase:
199 Data Structures :
200 -
201 -
202
203 Aglorithm Name:
204 Time Complexity:
205 Space Complexity:
206 Solution Technique:
207
208 */
209 int main() {
210
211
212     ReadAllObjectBalls();
213     PrintAllObjectBallInfo();
214
215
216
217     return 0;
218 }
```