Queens College, CUNY, Department of Computer Science

Object-oriented programming in C++

CSCI 211 / 611

Summer 2018

Instructor: Dr. Sateesh Mane

Midterm 1

sample programs

General information

- The questions in this exam do not involve problems of overflow.
- Solutions involving the writing of programs will not be judged if they work on a 64–bit instead of a 32–bit computer.
- **In all questions where you are asked to write code, you may assume the following:**
  1. All relevant header files are supplied to you.
  2. The statement "`using namespace std;`" is written for you (after the header files).

1 Material to be used in later questions

- Form a set of eight digits $(d_1, \ldots, d_8)$ as follows.
- **Take the 8 digits of your student id and define $(d_1, \ldots, d_8)$ as follows:**

$$\begin{aligned} d_1 &= \text{digit 1 of student id}\,, \\ d_2 &= \text{digit 2 of student id}\,, \\ d_3 &= \text{digit 3 of student id}\,, \\ d_4 &= \text{digit 4 of student id}\,, \\ d_5 &= \text{digit 5 of student id}\,, \\ d_6 &= \text{digit 6 of student id}\,, \\ d_7 &= \text{digit 7 of student id}\,, \\ d_8 &= \text{digit 8 of student id}\,. \end{aligned} \tag{1.1}$$

- For example if your student id is 23054611, then

$$\begin{aligned} d_1 &= 2\,, \\ d_2 &= 3\,, \\ d_3 &= 0\,, \\ d_4 &= 5\,, \\ d_5 &= 4\,, \\ d_6 &= 6\,, \\ d_7 &= 1\,, \\ d_8 &= 1\,. \end{aligned} \tag{1.2}$$

- *For some student ids, it is possible that some of the digits may be zero. It is also possible that some of the digits may be equal.* Do not worry.
- For the student id 11111111, all the digits are equal.
- For the student id 33330000, four digits are zero and the other four are all equal to 3.

2 Question 2

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

void comp1(int S, int K, int B)
{
  if (S < K < B)
    cout << "true" << endl;
  else
    cout << "false" << endl;
}

void comp2(int S, int K, int B)
{
  if (B > K > S)
    cout << "true" << endl;
  else
    cout << "false" << endl;
}

int main()
{
  comp1(100, 110, 120);             // (a)
  comp2(100, 110, 120);             // (b)

  comp1(100, 120, 110);             // (c)
  comp2(100, 120, 110);             // (d)

  comp1(120, 110, 100);             // (e)
  comp2(120, 110, 100);             // (f)
  return 0;
}
```

3 Question 3

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

double& func0(double &n)
{
  return ++n;
}

int main()
{
  double x0 = 6;
  double y0 = func0(x0);
  cout << "(a) " << x0 << "  " << y0 << endl;
  y0 = func0(x0);
  cout << "(b) " << x0 << "  " << y0 << endl;
  return 0;
}
```

3.1 Case 1

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

double func1(double n)
{
  double m = ++n;
  return m;
}

int main() {
  double x1 = d1+d7;   // digits 1 and 7 of student id (see Section 1)

  double y1 = func1(x1);
  cout << "(a) " << x1 << "  " << y1 << endl;
  y1 = func1(x1);
  cout << "(b) " << x1 << "  " << y1 << endl;

  return 0;
}
```

3.2 Case 2

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

double func2(double n)
{
  double m = n++;
  return m;
}

int main() {
  double x2 = (d2+d8)/2;   // digits 2 and 8 of student id (see Section 1)

  double y2 = func2(x2);
  cout << "(a) " << x2 << "  " << y2 << endl;
  y2 = func2(x2);
  cout << "(b) " << x2 << "  " << y2 << endl;

  return 0;
}
```

3.3 Case 3

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

double func3(double &n)
{
  double m = ++n;
  return m;
}

int main() {
  double x3 = (d3+d7)/3;   // digits 3 and 7 of student id (see Section 1)

  double y3 = func3(x3);
  cout << "(a) " << x3 << "  " << y3 << endl;
  y3 = func3(x3);
  cout << "(b) " << x3 << "  " << y3 << endl;

  return 0;
}
```

3.4 Case 4

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

double func4(double &n)
{
  double m = n++;
  return m;
}

int main() {
  double x4 = (d4+d8)/4;   // digits 4 and 8 of student id (see Section 1)

  double y4 = func4(x4);
  cout << "(a) " << x4 << "  " << y4 << endl;
  y4 = func4(x4);
  cout << "(b) " << x4 << "  " << y4 << endl;

  return 0;
}
```

3.5 Case 5

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

double& func5(double n)
{
  double m = n + 1;
  return m;
}

int main() {
  double x5 = (d5+d7)/5;   // digits 5 and 7 of student id (see Section 1)

  double y5 = func5(x5);
  cout << "(a) " << x5 << "  " << y5 << endl;
  y5 = func5(x5);
  cout << "(b) " << x5 << "  " << y5 << endl;

  return 0;
}
```

3.6 Case 6

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

double& func6(double &n)
{
  double m = n + 1;
  return m;
}

int main() {
  double x6 = (d6+d8)/6;   // digits 6 and 8 of student id (see Section 1)

  double y6 = func6(x6);
  cout << "(a) " << x6 << "  " << y6 << endl;
  y6 = func6(x6);
  cout << "(b) " << x6 << "  " << y6 << endl;

  return 0;
}
```

4 Question 4

4.1 Case 1

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

void sum_array1(int n, long array[], long sum)
{
  for (int i = 0; i < n; ++i) {
    sum = sum + array[i];
  }
}

int main()
{
  int n = 4;
  long sum = 0;
  long a[4];
  a[0] = -1;
  a[1] = 2;
  a[2] = 3;
  a[3] = 4;
  sum_array1(n, a, sum);
  cout << "(a) " << sum << endl;
  sum_array1(n, a, sum);
  cout << "(b) " << sum << endl;
  return 0;
}
```

4.2 Case 2

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

long sum_array2(int n, long array[])
{
  long sum;
  for (int i = 0; i < n; ++i) {
    sum = sum + array[i];
  }
  return sum;
}

int main()
{
  int n = 4;
  long sum = 0;
  long a[4];
  a[0] = -1;
  a[1] = 2;
  a[2] = 3;
  a[3] = 4;
  sum = sum_array2(n, a);
  cout << "(a) " << sum << endl;
  sum = sum_array2(n, a);
  cout << "(b) " << sum << endl;
  return 0;
}
```

4.3 Case 3

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

void sum_array3(int n, long array[], long & sum)
{
  for (int i = 0; i < n; ++i) {
    sum = sum + array[i];
  }
}

int main()
{
  int n = 4;
  long sum = 0;
  long a[4];
  a[0] = -1;
  a[1] = 2;
  a[2] = 3;
  a[3] = 4;
  sum_array3(n, a, sum);
  cout << "(a) " << sum << endl;
  sum_array3(n, a, sum);
  cout << "(b) " << sum << endl;
  return 0;
}
```

4.4 Case 4

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

long sum_array4(int n, long array[])
{
  long sum = 0;
  int i = 0;
  while (i < n) {
    sum = sum + array[i];
  }
  return sum;
}

int main()
{
  int n = 4;
  long sum = 0;
  long a[4];
  a[0] = -1;
  a[1] = 2;
  a[2] = 3;
  a[3] = 4;
  sum = sum_array4(n, a);
  cout << "(a) " << sum << endl;
  sum = sum_array4(n, a);
  cout << "(b) " << sum << endl;
  return 0;
}
```

4.5 Case 5

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

long sum_array5(int n, long array[])
{
  long sum = 0;
  while (n) {
    sum += array[--n];
  }
  return sum;
}

int main()
{
  int n = 4;
  long sum = 0;
  long a[4];
  a[0] = -1;
  a[1] = 2;
  a[2] = 3;
  a[3] = 4;
  sum = sum_array5(n, a);
  cout << "(a) " << sum << endl;
  sum = sum_array5(n, a);
  cout << "(b) " << sum << endl;
  return 0;
}
```