IIT Bombay

Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: An Example with Sequential and Conditional Execution

Quick Recap of Some Relevant Topics

IIT Bombay

- Structure of a simple C++ program
- Variables and type declarations
- Assignment statements
- Arithmetic and logical expressions
- Sequential execution of statements
- Conditional execution using “if … else …” and “switch … case…”
- Conditional expressions

Overview of This Lecture

- Putting it all together
- A simple, yet intelligent “fortune” program in C++

An intelligent “fortune” program

IIT Bombay

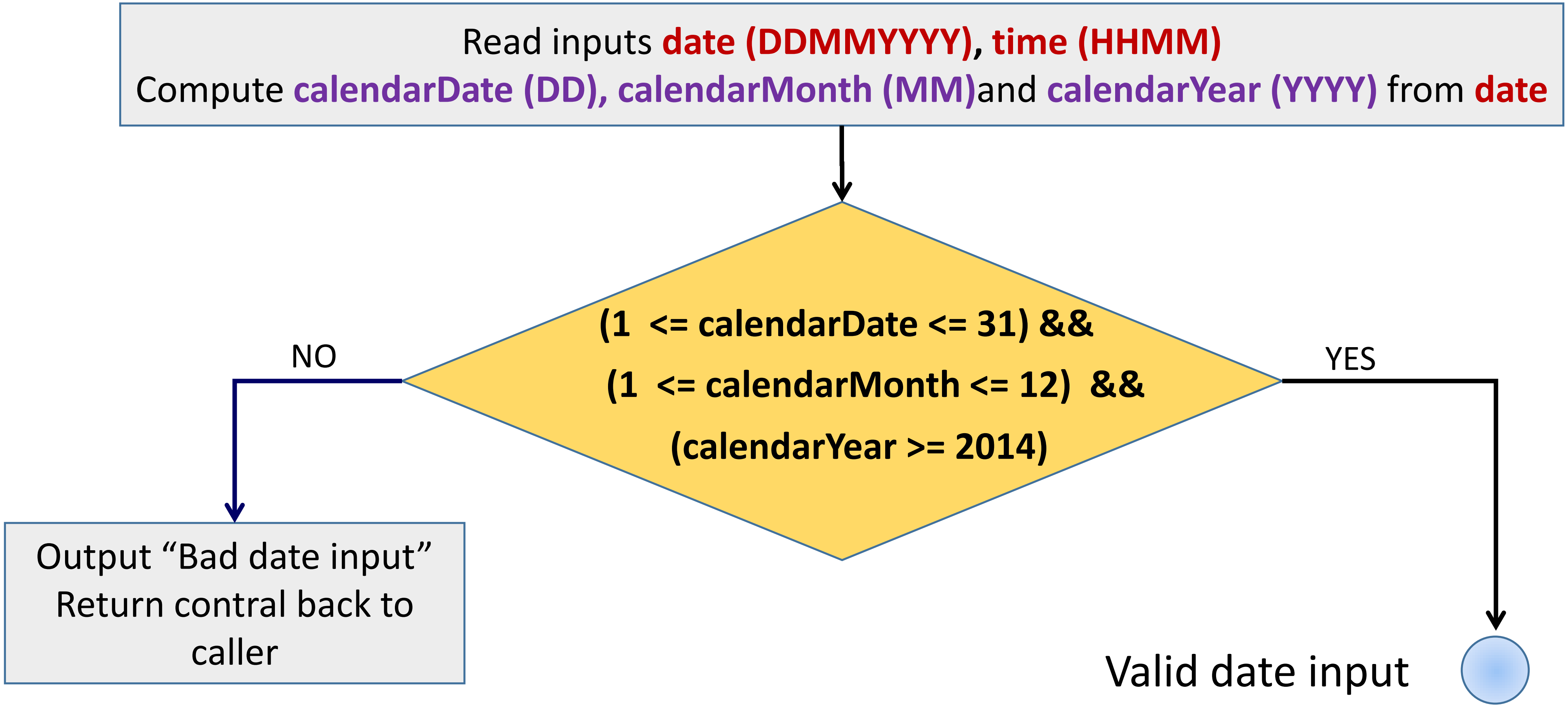
Given date (DDMMYYYY) and time (HHMM) as integers, Check for invalid date and time

If inputs are valid, output “Good morning”, “Good afternoon”, or “Good evening” depending on time of day

Output one of three pre-determined “fortune” messages

Flowchart for Checking Validity of Date

IIT Bombay



Read inputs **date (DDMMYYYY)**, **time (HHMM)**
Compute **calendarDate (DD), calendarMonth (MM)**and **calendarYear (YYYY)** from **date**

(1 <= calendarDate <= 31) &&
(1 <= calendarMonth <= 12) &&
(calendarYear >= 2014)

Output “Bad date input”
Return contral back to caller

Valid date input

Checking Validity of Date in C++

```
int main() {
  int date, time, calendarYear, calendarDateAndTime, calendarDate, calendarTime;
  int hour, minute, hash;  // To be used in later part of program
  cout << “Give date (DDMMYYYY) and time (HHMM): “;
  cin >> date >> time;  // Suppose date is 22072014 and time is 1345
  calendarYear = date % 10000;   // 22072014 % 10000 = 2014
  calendarDateAndMonth = date / 10000;  // 22072014/10000 = 2207
  calendarMonth = calendarDateAndMonth % 100; // 2207 % 100 = 7
  calendarDate = calendarDateAndMonth / 100; // 2207/100 = 22
  if ((calendarDate > 31) || (calendarDate < 1) || (calendarMonth < 1) || (calendarMonth > 12)
     ||(calendarYear < 2014)) {
      cout << “Bad date input.” << endl;  return -1;
  }
    //  Further code comes here
 return 0;
}
```

Checking Validity of Date in C++

IIT Bombay

```
int main() {
  int date, time, calendarYear, calendarDateAndTime, calendarDate, calendarTime;
  int hour, minute, hash; // To be used in later part of program
  cout << “Give date (DDMMYYYY) and time (HHMM): “;
  cin >> date >> time;  // Suppose date is 22072014 and time is 1345
  calendarYear = date % 10000;   // 22072014 % 10000 = 2014
  calendarDateAndMonth = date / 10000;  // 22072014/10000 = 2207
  calendarMonth = calendarDateAndMonth % 100; // 2207 % 100 = 7
  calendarDate = calendarDateAndMonth / 100; // 2207/100 = 22
  if ((calendarDate > 31) || (calendarDate < 1) || (calendarMonth < 1) || (calendarMonth > 12)
      ||(calendarYear < 2014)) {
       cout << “Bad date input.” << endl;  return -1;
  }
    //  Further code comes here
 return 0;
}
```

Checking Validity of Date in C++

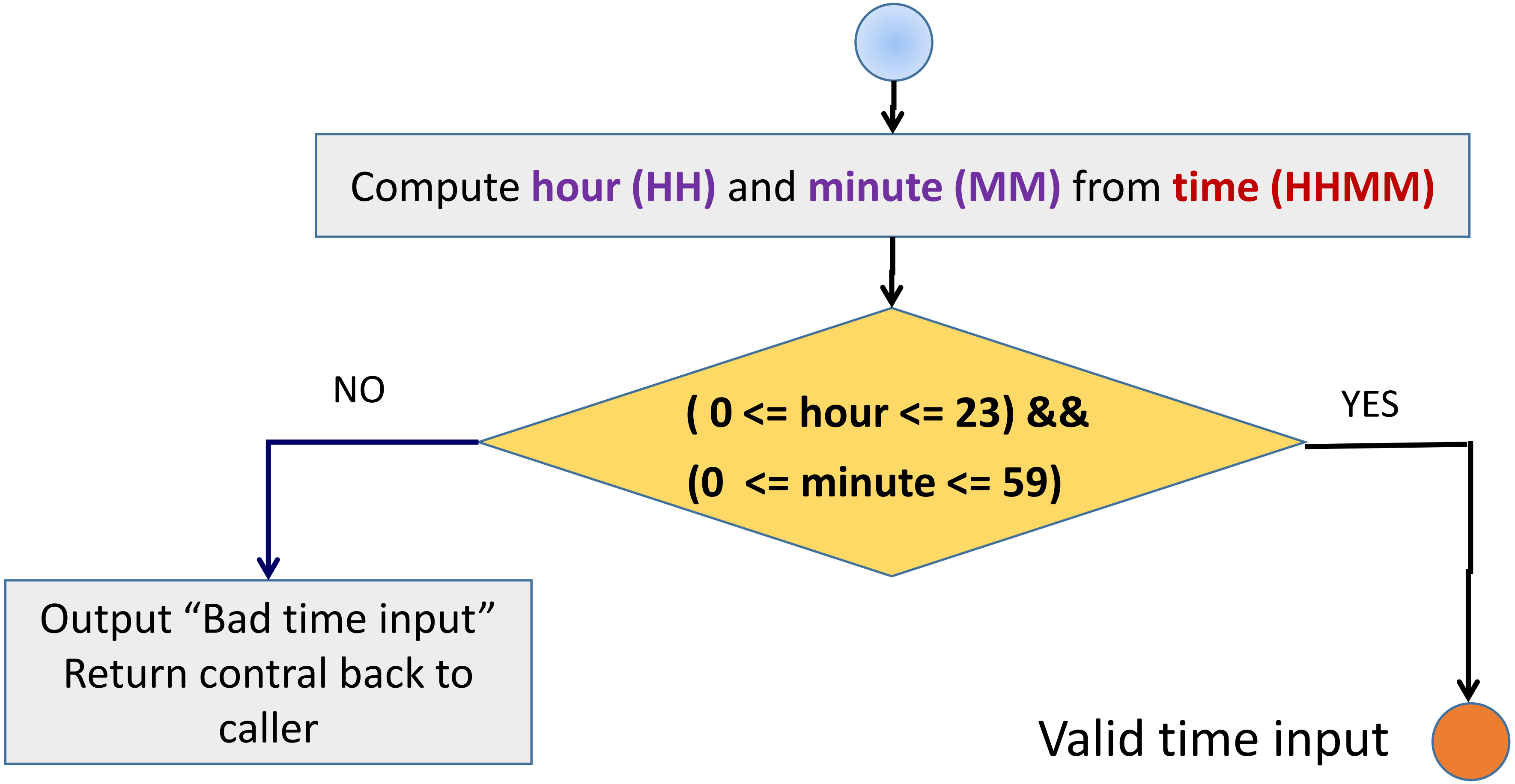
IIT Bombay

```
int main() {
  int date, time, calendarYear, calendarDateAndTime, calendarDate, calendarTime;
  int hour, minute, hash; // To be used in later part of program
  cout << “Give date (DDMMYYYY) and time (HHMM): “;
  cin >> date >> time;  // Suppose date is 22072014 and time is 1345
  calendarYear = date % 10000;   // 22072014 % 10000 = 2014
  calendarDateAndMonth = date / 10000;  // 22072014/10000 = 2207
  calendarMonth = calendarDateAndMonth % 100; // 2207 % 100 = 7
  calendarDate = calendarDateAndMonth / 100; // 2207/100 = 22
  if ((calendarDate > 31) || (calendarDate < 1) || (calendarMonth < 1) || (calendarMonth > 12)
     ||(calendarYear < 2014)) {
      cout << “Bad date input.” << endl;  return -1;
  }
   //  Further code comes here
 return 0;
}
```

Checking Validity of Date in C++

```
int main() {
  int date, time, calendarYear, calendarDateAndTime, calendarDate, calendarTime;
  int hour, minute, hash; // To be used in later part of program
  cout << “Give date (DDMMYYYY) and time (HHMM): “;
  cin >> date >> time;  // Suppose date is 22072014 and time is 1345
  calendarYear = date % 10000;   // 22072014 % 10000 = 2014
  calendarDateAndMonth = date / 10000;  // 22072014/10000 = 2207
  calendarMonth = calendarDateAndMonth % 100; // 2207 % 100 = 7
  calendarDate = calendarDateAndMonth / 100; // 2207/100 = 22
  if ((calendarDate > 31) || (calendarDate < 1) || (calendarMonth < 1) || (calendarMonth > 12)
     ||(calendarYear < 2014)) {
      cout << “Bad date input.” << endl;  return -1;
  }
    //  Further code comes here
 return 0;
}
```

Flowchart for Checking Validity of Time


Compute **hour (HH)** and **minute (MM)** from **time (HHMM)**

(0 <= hour <= 23) &&
(0 <= minute <= 59)

NO

YES

Output “Bad time input”
Return contral back to
caller

Valid time input

Checking Validity of Time in C++

```
int main() {
  … Declarations and code for checking validity of date …

  // Suppose time = 1345
  hour = time/100;  // 1345/100 = 13
  minute = time % 100;  // 1345 % 100 = 45

  if ((hour < 0) || (hour > 23) || (minute < 0) || (minute > 59)) {
    cout << “Bad time input.” << endl; return -1;
  }
  // Further code comes here
 return 0;
}
```

Printing Time-Dependent Greeting in C++

```
int main() {
 … Declarations and code for checking validity of date and time …
 // Print greeting
 if ((6 <= hour) && (hour < 12)) {
   cout << “Good morning!” << endl;
 }
 else {
        if ((12 <= hour) && (hour <= 18)) {
           cout << “Good afternoon!” << endl;
         }
         else {cout << “Good evening!” << endl;}
  }
 // Further code comes here
 return 0;
}
```

Printing “fortune” message in C++

IIT Bombay

```
int main() {
 … Declarations and code for checking validity of date and time …
 … Code for printing greeting …

  hash = (date + time) % 3;  // Get a value in {0, 1, 2}
  switch (hash) {
    case 0: cout << “Time and tide wait for none.” << endl;  break;
    case 1: cout << “The pen is mightier than the sword.” << endl; break;
    default: cout << “Where there is a will, there is a way.” << endl;
  }
  return 0;
}
```

Summary

- A simple, yet interesting program that uses
  - Integer variables
  - Assignment statement with arithmetic expressions
  - Logical expressions
  - Sequential execution
  - Conditional execution using “if … else …” statements

    Nested “if … else … “ statements
  - Condition execution using “switch … case …” statements
  - “cin” and “cout”