Computer Programming

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

Session: Quiz and Practice Questions on Classes – Part 1

Recap Quiz

IIT Bombay

Q1. A private member function of a structure can be invoked from

(A) Another member function of the same structure

(B) A member function of another structure

(C) Any ordinary (non-member) function in the program

(D) The "main" function in the program

Recap Quiz

IIT Bombay

Q2. Consider the class definition

```
class CL { private: int a;  void f1() { … };
           public:  int b;   void f2() { … }; };
```

Which of the following is/are true of class CL ?

A. "a" and "b" can be used in the body of f2

B. "b" but not "a" can be used in the body of f2

C. "a" cannot be used in the body of f1

D. "b" cannot be used in the body of f1

Recap Quiz

IIT Bombay

Q3. By default, everything in a struct is assumed to be __B1__ and everything in a class is assumed to be __B2__.

A. B1: private, B2: public

B. B1: private, B2: private

C. B1: public, B2: private

D. B1: public, B2: public

Recap Quiz

Q4. Accessor and mutator functions can be used to

A. Read and write only private data members

B. Read and write any data member

C. Hide internal representation details of data members

D. Expose internal representation details of data members

Recap Quiz

IIT Bombay

Q5. A class CL has 3 constructor functions. Which of the following must be true?

A. The constructor functions must have different names

B. The constructor functions must have different return types

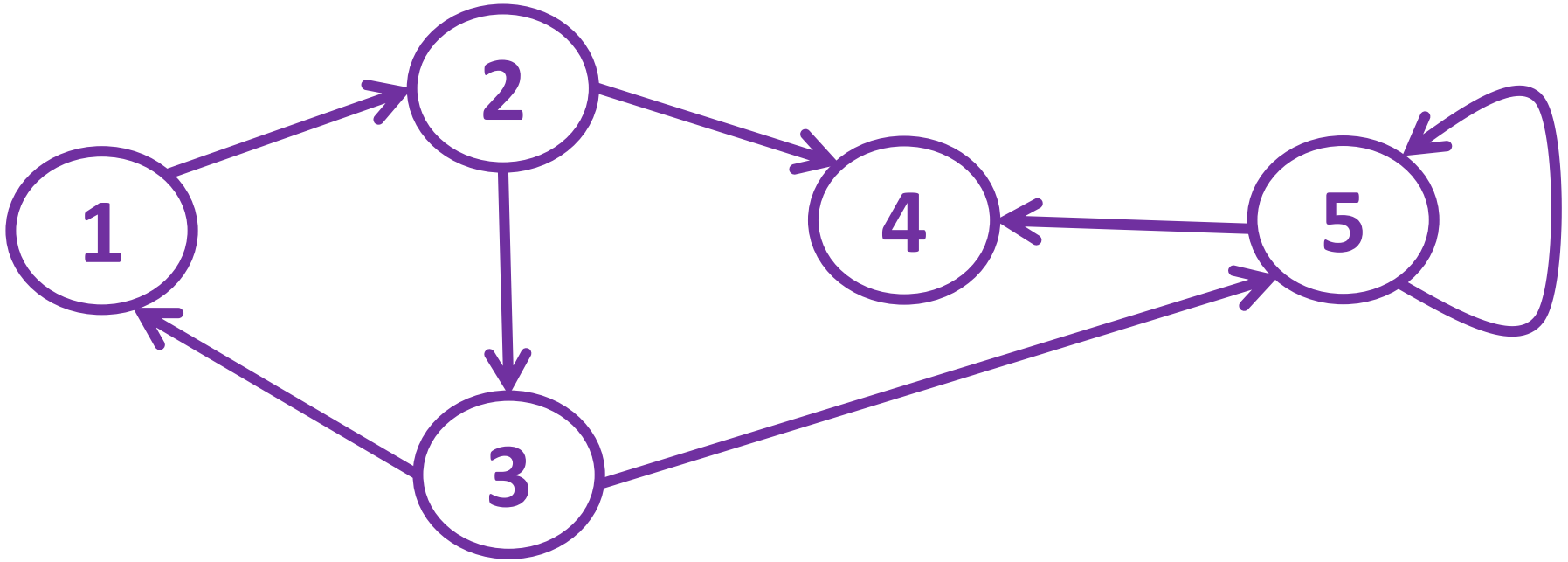
C. The constructor functions must have different lists of parameter types

D. All of the above

Practice Question 1

IIT Bombay

We studied about directed graphs in last class.



Practice Question 1 (Recap from last class)

IIT Bombay

- **Nodes must be represented using a structure**

  struct myNode { … };

- **Assume all nodes in the graph are stored in an array named "nodes". Id of a node is its index in the array.**

Practice Question 1A

IIT Bombay

We will use the following structure to represent a node

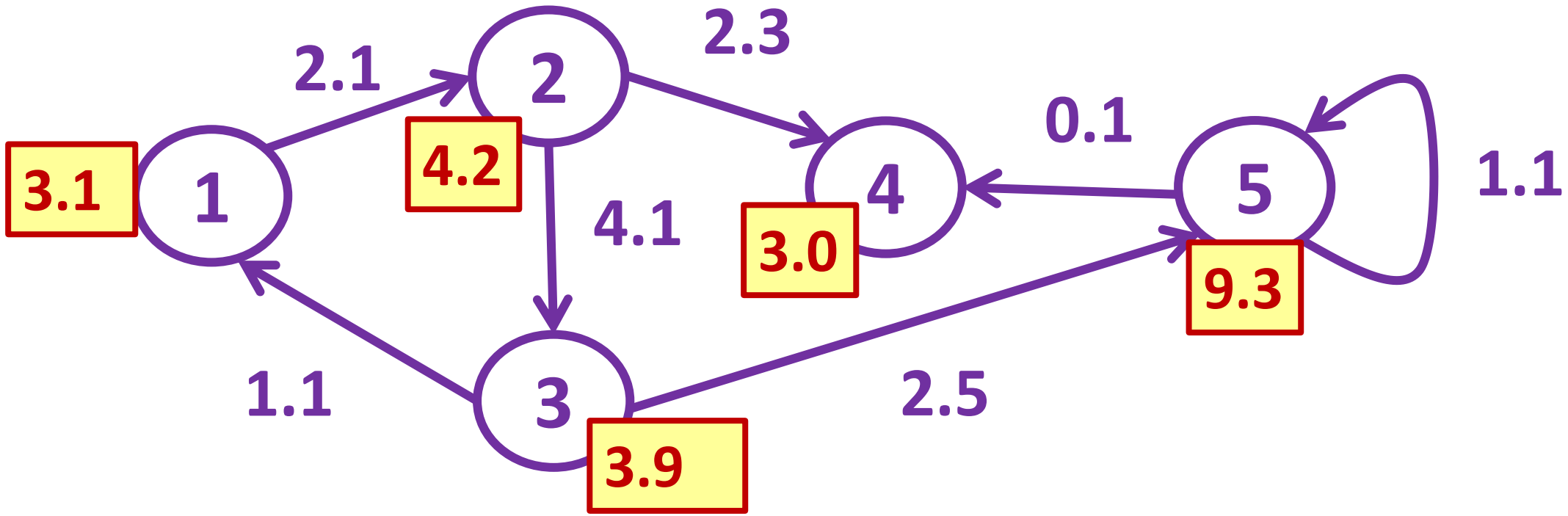
```
struct myNode {
    int id;
    LinkedNodes *outgoing;
    LinkedNodes *incoming;
}
```

```
struct LinkedNodes {
   int nodeId;
   LinkedNodes *next;
}
```

Practice Question 1

IIT Bombay

- **We now want each node and each edge to have a weight (of type float)**



Practice Question 1 (Recap from last class)

- Modify the myNode and LinkedNodes structure definitions (define these to be classes) to be able to represent directed graphs with weighted nodes and edges
- Ensure that the weight of a node cannot be accessed directly from the main program, and similarly for the weight of an edge, its incoming and outgoing edges.
- Feel free to add appropriate member functions with appropriate access control

Practice Question 1A

IIT Bombay

```
int main () {
  int numNodes;
  cout << "Give no. of nodes: ";  cin >> numNodes;
  myNode *nodes = new myNode[numNodes];
  if (nodes == NULL) {
     cout << "Memory allocation failure." << endl;
     return -1;
  }
  else { newInitNodes(nodes, numNodes); }
```

Can ask for weights of nodes

(continued on next slide …)

Practice Question 1A

IIT Bombay

```
int  startEdge, endEdge;  float edgeWt;
while (true) {
    // Reading in edges, one at a time
    cout << "Give start of edge (-1 to quit): ";
    cin >> startEdge;  if (startEdge == -1) break;
    cout << "Give end of edge (-1 to quit): ";
    cin >> endEdge;  if (endEdge == -1) break;
    cout << "Weight of edge: "; cin >> edgeWt;
    newAddEdge(nodes, startEdge, endEdge, edgeWt);
}   (continued on next slide …)
```

Practice Question 1A

IIT Bombay

```
// Printing adjacent nodes of every node
for (int i = 0; i < numNodes; i++) {
  cout << "Metric for node " << i << ": "
  // Metric (sum of in edge wts – sum of out edge wts)
  //                          node wt
  nodes[i].printMetric();
}
return 0;
}
```

Practice Question 1A

IIT Bombay

Write the functions

```
void newinitNodes(myNodes *nodes,
                  int numNodes);
void newaddEdge (myNodes *nodes,
                 int start, int end, float wt);
void printMetric();
```