



# Indian Institute of Technology Kharagpur

## End-Spring Semester Examination 2022-23

Date of Examination: Apr, 2023

Duration: 3 Hours

Subject No.: CS20006/CS20202

Subject: Software Engineering

Department/Center/School: Computer Science

Credits: 3

Full marks: 70

Name: \_\_\_\_\_

Roll Number: \_\_\_\_\_

### Instructions

- Please write your name and roll number above before attempting any solution.
- All questions are compulsory. Be brief and precise. Mysterious or unsupported answers will not receive full marks.
- Write your answers in the space provided in this question paper booklet.
- Answer to each question should be strictly within the space provided for that question. If the space provided for the answer is not sufficient, the rest of the answer may be written in the Extra Space provided after all questions. This should be clearly indicated in the space provided, failing which, the extra answer may not be checked.
- Use of electronic calculators only is permitted. No extra resources viz. graph papers, log-tables, trigonometric tables would be required.

---

Question:	1	2	3	4	5	6	Total
Points:	15	10	10	10	10	15	70
Score:							

---

1. Write proper justifications to your answers.
- (a) (2 points) Consider the following code segment.

```
1 #include <iostream>
2 using namespace std;
3
4 class bird{
5     public:
6         void fly(){ cout << "bird::fly()"; }
7         void fly(int height){ cout << "bird::fly(int)"; }
8 };
9
10 class crow : public bird{
11     public:
12         void fly(){ cout << "crow::fly()"; }
13         void fly(int height){ cout << "crow::fly(int)"; }
14 };
15
16 int main() {
17     crow c;
18     _____;
19     return 0;
20 }
```

Fill in the blank at line 18 such that the output is `bird::fly()`. The options are

- (a) `c.fly()` (b) `bird::fly()` (c) `c.bird::fly()` (d) `bird::c.fly()`

**Solution:** (c) `c.bird::fly()`

We can access a base class function using the scope resolution operator even if it is hidden by a derived class function. Hence, c) is the correct option.

- (b) (2 points) Consider the class definition given below.

```
1 class base{
2     public:
3         void fun1(int);
4         void fun2(char*);
5         void fun3();
6 };
7
8 class derived : public base{
9     public:
10        void fun1();
11        void fun1(int);
12        void fun3();
13        void fun4(char*);
14 };
```

Which of the following statements is/are true?

- a) `derived::fun1()` and `derived::fun1(int)` override `base::fun1()`  
b) `derived::fun1()` overloads and `derived::fun1(int)` overrides `base::fun1(int)`

- c) `derived::fun4(char*)` overrides `derived::fun2(char*)` and `derived::fun3()` overrides `derived::fun3()`
- d) `derived::fun3()` overrides `base::fun3()` and `derived::fun1(int)` overrides `base::fun1(int)`

**Solution:** b), d)

- (c) (2 points) Consider the code segment given below.

```
1 #include <iostream>
2 using namespace std;
3
4 class Person{
5     string name;
6     public:
7         Person(string n) : name(n){ }
8         void displayname(){ cout << name << " "; }
9 };
10
11 class Student : public Person{
12     int roll;
13     public:
14         Student(string n, int r) : Person(n), roll(r) { }
15         void displayroll(){ cout << roll << " "; }
16 };
17
18 int main(){
19     Student s1("Arup", 101);
20     Person *p = &s1;
21     p->displayname();
22     p->displayroll();
23     return 0;
24 }
```

What will be the output/error?

**Solution:** Compilation error at line 22: class Person has no member named 'displayroll'.

(d) (3 points) Consider the code segment given below.

```
1 #include <iostream>
2 using namespace std;
3
4 class A{
5     public:
6         _____;
7 };
8
9 class B : public A{
10     public:
11         void fun(){ cout << "B::fun" << endl; }
12 };
13
14 int main(){
15     A *t = new B;
16     t->fun();
17     return 0;
18 }
```

Fill in the blank at line 6 such that the program gives output as - B::fun. The options are: (a) void fun() = 0 (b) virtual void fun() (c) virtual void fun() = 0. Give proper justification for each of the three options why one is a viable option or not.

**Solution:** c)

Option a) declares a non-virtual function, which requires function definition (or function body). Thus, a) is incorrect. Option b) is not a pure virtual function, which also needs the function definition. Option c) is a pure virtual function, which does not require any definition. So it is the correct option.

(e) (3 points) Consider the code segment given below.

```
1 #include <iostream>
2 using namespace std;
3
4 class myClass1{
5     public:
6         virtual void A(){ cout << "myClass1::A" << endl; }
7         virtual void B() = 0;
8         void C(){ cout << "myClass1::C" << endl; }
9 };
10
11 class myClass2 : public myClass1{
12     public:
13         void A(){ cout << "myClass2::A()" << endl; }
14         virtual void B(){ cout << "myClass2::B()" << endl; }
15         void C(){ cout << "myClass2::C" << endl; }
16 };
17
```

```
18 int main() {
19     myClass2 c2;
20     myClass1 *m1 = &c2;
21     myClass1 &m2 = c2;
22
23     m1->A();
24     m2.B();
25     m2.C();
26
27     return 0;
28 }
```

What will be the result of lines 23, 24 and 25? Identify if the respective function calls in these lines are bound statically or dynamically.

**Solution:** myClass2::A()  
myClass2::B()  
myClass1::C

As the functions A() and B() are virtual in class myClass1, the calls m1->A() and m2.B() with a pointer type or reference type would be dynamic bindings. However, function C() is non-virtual. Hence, the call m2.C() would be static binding.

(f) (3 points) Consider the code segment given below.

```
1 #include <iostream>
2 using namespace std;
3
4 template<typename T> T sum(const T& x, const T& y) {
5     return x + y;
6 }
7
8 int main() {
9     cout << _____;
10    return 0;
11 }
```

Identify the incorrect statement(s) about the above program with proper justifications.

- a) Filling the blank at Line 9 with sum(10, 20) statement prints 30
- b) Filling the blank at Line 9 with sum(10, 20.0) statement prints 30.0
- c) Filling the blank at Line 9 with sum(1.2, 2.14) statement prints 3.34
- d) Filling the blank at Line 9 with sum(1.2f, 2.14) statement prints 3.34

**Solution:** b), d)

In b), the first parameter is int and the second parameter is double. Similarly, in d), the first parameter is float and the second parameter is double. So in both the case, the instantiation of T is ambiguous.

2. (a) (5 points) Consider the following code segment.

```
1 #include <iostream>
2 using namespace std;
3 class point{
4     int x, y;
5     public:
6         point(int x_ = 0, int y_ = 0) : x(x_), y(y_){cout << "ctor ";}
7         point(point &p) : x(p.x), y(p.y){cout << "c-ctor ";}
8         point& operator=(point p)
9             {x = p.x; y = p.y; cout << "c-assign "; return *this;}
10 };
11 int main() {
12     point p1(10, 20);
13     point p2 = p1;
14     point *pt;
15     point p3;
16     p3 = p2;
17     return 0;
18 }
```

What will be the output/error? Briefly justify your answer.

**Solution:** ctor c-ctor ctor c-ctor c-assign

The statement `point p1(10, 20);` invokes the constructor and prints `ctor`. The statement `point p2 = p1;` invokes the copy constructor and prints `c-ctor`. The statement `point *pt;` does not create an object. The statement `point p3;` invokes the constructor and prints `ctor`. The statement `p3 = p2;` calls the copy assignment operator. Since it passes `p2` by value, it invokes copy constructor and prints `c-ctor`. Then executes the body of the copy assignment operator and prints `c-assign`.

- (b) (5 points) Consider the following program. Fill in Line-8, Line-12 and Line-16 with appropriate inheritance statements such that it will satisfy the given test cases. Properly justify your answer.

```
1 #include <iostream>
2 using namespace std;
3 class Base {
4     public:
5         Base(int x = 0) { cout << x << " "; }
6         void fun(int i){ cout << i << " "; }
7 };
```

```

8 class D1 : _____ {//Inherit from class Base
9     public:
10         D1(int x = 0) : Base(x+1) { fun(x+1); }
11 };
12 class D2 : _____ {//Inherit from class Base
13 public:
14         D2(int x = 0) : Base(x+2) { fun(x+2); }
15 };
16 class DD : _____ {//Inherit from class D1 & D2
17     public:
18         DD(int i = 0) : D1(i*2), D2(++i) { fun(i); }
19 };
20 int main() {
21     int i;
22     cin >> i;
23     DD obj(i);
24     return 0;
25 }

```

Test Case I

Input: 2

Output: 0 5 5 3

Test Case II

Input: 3

Output: 0 7 6 4

Test Case III

Input: 5

Output: 0 11 8 6

**Solution:** LINE-8: virtual public Base **or** public virtual Base

LINE-12: virtual public Base **or** public virtual Base

LINE-16: public D1, public D2

From the all test-cases, we can see that the default constructor for the root class Base is called. And it is always called only once. Hence, the classes D1 and D2 used virtual inheritance. So the LINE-8 and LINE-12 need to be filled by public virtual Base or virtual public Base. Class DD should be inherited from class D1 and D2 according to the question. class D1 is invoked before class D2. Hence, the LINE-16 need to be filled by public D1, public D2.

3. (a) (2 points) Consider the following code segment.

```

1 #include <iostream>
2 using namespace std;
3 class myClass {
4     int a;
5 public:
6     myClass(int _a=0) : a(_a) { }
7     _____
8 };

```

```
9 void display(myClass &m) {  
10     cout << m.a;  
11 }  
12  
13 int main() {  
14     myClass m(5);  
15     display(m);  
16     return 0;  
17 }
```

Which option(s) at Line-7 will result in printing 5? Justify your answer. The options are:

- a) void friend display(myClass&);
- b) static void display(myClass&);
- c) friend void display(myClass&);
- d) void const display(myClass&);

**Solution:** a), c)

The global function display() accesses the private data member of class myClass. This can only be done when the function is declared as friend of the class. Hence, correct options are a) and c).

(b) (2 points) Consider the following code segment.

```
1 #include <iostream>  
2 using namespace std;  
3 namespace name1 {  
4     int a = 1;  
5     int b = 2;  
6 }  
7 namespace name2 {  
8     int c = 3;  
9     int d = 4;  
10 }  
11 using namespace name1;  
12 int main() {  
13     cout << a << endl;  
14     cout << name1::b << endl;  
15     cout << name2::c << endl;  
16     cout << d << endl;  
17     return 0;  
18 }
```

Which line/s inside main() will give error and why?

**Solution:** Line 16

The variable d is declared within namespace name2 and is not accessible within the scope of main function because the namespace name2 is not used. Hence, Line 16 will give an error.



- (c) (6 points) Consider the following program. Fill in the blanks as per the instructions below:
- at LINE-1 with appropriate keyword to declare the Database pointer variable,
  - at LINE-2 to complete the header for function getInstance(int i), and
  - at LINE-3 with appropriate keyword to complete ins variable initialization
- such that it will satisfy the given test cases. Justify your answer.

```

1 #include <iostream>
2 using namespace std;
3 class Database {
4     int data;
5     _____ Database *ins;           // LINE-1
6     Database(int i) : data(i) { }
7 public:
8     int get() { return data; }
9     _____ getInstance(int i) {     // LINE-2
10         if(!ins)
11             ins = new Database(i);
12         return ins;}
13 };
14 _____ Database::ins = 0;           // LINE-3
15
16 int main() {
17     int n, i;
18     cin >> n;
19     int a[n];
20     for(i=1;i<=n;i++)
21         cin >> a[i];
22     for(i=1;i<=n;i++) {
23         Database *ins = Database::getInstance(a[i]);
24         cout << ins->get() << " "; }
25     return 0;
26 }
```

Test Case I		Test Case II		Test Case III	
Input:	Output:	Input:	Output:	Input:	Output:
2		3		2	
3 4	3 3	2 4 6	2 2 2	5 7	5 5

**Solution:** LINE-1: static, LINE-2: static Database\*

LINE-3: Database\*

As per the test cases given, the pointer variable ins is initialized only once at the time of first object creation in the main function. The same instance is used for all objects. It can be done only when the LINE-1 is filled as static.

The function getInstance() is being called using the class name. It can only be done when the function is declared static. Hence, LINE-2 will be filled as static Database\*.

The initialization of static variable at LINE-3 can be done as

Database\* Database::ins = 0;

4. (a) (2 points) Answer in 1 sentence:

1. What do nodes and edges of a class diagram denote?

**Solution:** classes and relations between classes

2. What is the difference between weak aggregation and strong aggregation?

**Solution:**

(b) (6 points) Write the appropriate relations between the classes with reasons between each of the classes in the following cases. If multiple types of relations are possible, write the most specific one. Also, draw the appropriate connector between the classes with the appropriate arity at each end.

1. In the context of a Leave Management System, write the relationship between an Executive (a low-level employee, who is given tasks by a Manager) and a Manager (responsible for assigning tasks to executives, and also approving their leaves).

**Solution:** Specialization

Drawing of the connector:

**Solution:**

2. In the context of the Library Management System, write the relationship between the Catalogue (a class for storing the list of books in a library) and the Library (a class that stores all the information about the library).

**Solution:**

Weak aggregation, Has\_a

Drawing of the connector:

**Solution:**

3. In the context of Library Management System, write the relationship between a Book (a class storing all information about a book, e.g. name, year of publishing, etc.) and an Author (a class storing all the information about an author, e.g. name, date of birth, etc.).

**Solution:** Association, many to many

Drawing of the connector:

**Solution:**

- (c) (2 points) Answer the following questions in 1 sentence:

1. What are the nodes and edges of a Use-case diagram?

**Solution:** Nodes: Use cases and actors, edges connections between actors and use cases, and associations between use cases.

2. What is the difference between <include> and <extend> relations?

**Solution:**

5. (a) (4 points) Consider the following scenarios for the development of software. Write the name of the most appropriate SDLC model to be followed, with a 1-sentence reason.
1. A programming assignment is given in an undergraduate course where the requirements are defined clearly at the beginning, and evaluation of the assignment is performed one time after the submission.

**Solution:** Waterfall model.

2. A web portal needs to be developed for a small local store to take its business online. The specifications are very vague and small. The time required for the development should be low. Specifications may change with time. A detailed design document is not needed for the developed software.

**Solution:** Rapid Application Development.  
No marks for scrum or agile since customer cannot be part of team.

3. A Govt. organization issues a contract to a private company for developing a payment stack like the UPI. There are experts in the Govt. who want to be involved in the development and testing process. The development and a strict timeline for releases, and a prioritization of the use-cases which need to be developed first.

**Solution:** Agile - scrum or extreme programming.  
No marks for RAD since high quality code is needed.  
No marks for spiral models since there is a strict deadline.

4. A medical equipment company wants a feature to be added to one of the machines sold

by the company (say a Red Blood Cell count feature to be added to a digital microscope). The requirement is clear and constant. The team consists of Biotechnology engineers, who may not be the best programmers. However, the quality of the software should be exceptional since the application is critical. Cost of development (in terms of man-hours) can be high.

**Solution:** V-model.

No marks for spiral since it is not an open ended multi-feature project.

No marks for Scrum or XP since low quality developers are part of team.

No marks for TDD model since tester and developer are the same person. Here Biotechnology engineers can be testers but not developers.

(b) (1 point) Write the 3 main steps for test-driven-development methodology.

**Solution:** Red (adding new test cases until they fail), Green (making the code pass the test cases), and Refactor (make the overall structure of the code better).

(c) (5 points) Consider the following C++ code:

```
1 #include <iostream>
2 #include <algorithm>
3 #include <vector>
4 using namespace std;
5
6 class Adjdiff {
7     double preventry;
8     bool prevvalid;
9     public:
10    Adjdiff() : prevvalid(false) {}
11    vector<double> diffv;
12    void operator () (double d) {
13        double currdiff;
14        if (!prevvalid) {
15            prevvalid=true;
16            preventry=d;
17        }
18        else {
19            currdiff = d - preventry;
20            preventry = d;
21            diffv.push_back(currdiff);
22        }
23    }
24 };
25
```

```
26 int main() {  
27     vector<double> a= {1,2,3,4,5,6,7,8};  
28     Adjdiff ad;  
29     ad = for_each(a.begin(),a.end(),ad);  
30     vector<double>::iterator it;  
31     _____  
32     _____  
33     cout << endl;  
34 }
```

1. (3 points) Write the code using iterator `it` for printing the value of the vector `ad.diffv` (Blank lines 31 and 32).

**Solution:**

```
for(it = ad.diffv.begin(); it != ad.diffv.end(); it++)  
    cout << *it << " ";
```

2. (2 points) Write the output of the completed version of the above code (i.e. what will be the value(s) stored in the vector `ad.diffv`).

**Solution:**

```
1 1 1 1 1 1 1
```

6. (a) (5 points) In the context of Black-box testing, write the test cases for a program to calculate the mean and standard deviation of an input array. The inputs to the program are:
- A: An array of floats with each number in range  $[-1e10, 1e10]$ .
  - n: The number of elements in the input array should be between  $[0, 1000]$ .
- Both mean and standard deviation in case of zero inputs are 0.

Write at most 5 test cases covering as many equivalence classes as possible. For each test case, clearly write the input, and the testing scenario it covers.

**Solution:**

- Maximum number of input with maximum value (Overflow while calculating mean)
- Half the inputs with maximum value and half the inputs with minimum value, total number of inputs is maximum. (Overflow in calculating standard deviation).
- zero elements in the input.
- one element in input. (zero standard deviation case).
- a small number of inputs with the some randomly generated number

1 mark deducted if any of the inputs is missing.

1 mark deducted if no overflow checks

1 mark deducted if no boundary checks i.e. input is outside the limit

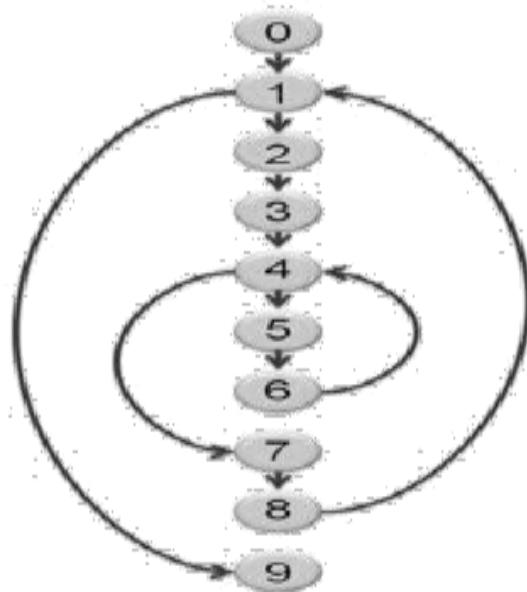
(b) (10 points) Consider the following pseudocode:

```

0: InsertionSort (A, n)
1:   for i = 2 to n {
2:     key = A[i]
3:     j=i-1
4:     while (j > 0) and (A [j] > key) {
5:       A[j+1]=A[j]
6:       j = j- 1
7:     }
8:     A [j+1] = key
9:   }

```

1. (4 points) Draw the control flow graph for the above program.

**Solution:**

2. (1 point) Calculate the cyclomatic complexity for the above control flow graph.

**Solution:**  $V(G) = E - N + 2$ ,  $E = 11$ ,  $N = 10$ ,  $V(G) = 3$

3. Consider the following test cases:

A = {1, 2, 3}, n=3

A= {3, 2, 1}, n=3

- (1 point) Calculate the statement coverage of the above set.

**Solution:** 10 / 10

No marks given for those who have reported per test-case coverage.

- (2 points) Calculate the branch/decision coverage of the above set.

**Solution:** 4 / 4

Branches: 1-2, 1-9, 4-5, 5-7

- (2 points) Calculate the path coverage of the above set.

**Solution:** 2 / 3

LIPs:

0-1-9

0-1-2-3-4-7-8-1-9

0-1-2-3-4-5-6-4-7-8-1-9



**Extra Sheets**





