Started on	Friday, 8 April 2022, 8:01 AM
State	Finished
Completed on	Friday, 8 April 2022, 9:00 AM
Time taken	58 mins 9 secs
Grade	Not yet graded

Question 1
Complete
Marked out of 8.00
Flag question

Consider the following program for calculation of correlation coefficient. Fill in the blank codes such that the code avoids division by zero in the case where one of the variances is zero. It should print the message `Variance is zero!`. [Marks 8]

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

float correlationCoefficient(int X[], int Y[], int n) {
    int sum_X = 0, sum_Y = 0, sum_XY = 0;
    int sqSum_X = 0, sqSum_Y = 0;
    for (int i = 0; i < n; i++)    {
        sum_X = sum_X + X[i]; sum_Y = sum_Y + Y[i];
        sum_XY = sum_XY + X[i] * Y[i];
        sqSum_X = sqSum_X + X[i] * X[i];
        sqSum_Y = sqSum_Y + Y[i] * Y[i];
    }
    if( ______BLANK_1______)
    throw new overflow_error("Variance is zero!");
    float corr = (float)(n * sum_XY - sum_X * sum_Y) /
    sqrt((n * sqSum_X - sum_X * sum_X) *
    (n * sqSum_Y - sum_Y * sum_Y));
    return corr;
}

int main() {
    int X[] = {15, 18, 21, 24, 27};
    int Y[] = {25, 25, 25, 25, 25};
    int n = sizeof(X)/sizeof(X[0]);
  try{
    cout<<correlationCoefficient(X, Y, n);
  }
  catch(____BLANK_2_____) {
     ____BLANK_3____
  }
    return 0;
}
```

Blank_1 :
(n*sqSum_X - sum_X * sum_X == 0) || (n*sqSum_Y - sum_Y * sum_Y == 0)

Blank_2 :
overflow_error &e

Blank_3 :
cout << e.what();

Question 2
Correct
Mark 5.00 out of 5.00
Flag question

What will be the output string printed by the program below when each of when it is executed after uncommenting each of the 4 `Throw lines`, one at a time ? [Marks 5]

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

class expA : public exception {
};

class expB : public expA {
};

int main() {
try {
//throw domain_error("domain_error");  // Throw Line-1
//throw 127;                           // Throw Line-2
//throw expB();                        // Throw Line-3
//throw range_error("range_error");    // Throw Line-4
}
catch (logic_error&) { // Catch Line-1
cout << "caught logic_error" << endl;
}
catch (expA&) { // Catch Line-2
cout << "caught expA" << endl;
}
catch (expB&) { // Catch Line-2
cout << "caught expB" << endl;
}
catch (exception&) { // Catch Line-3
cout << "caught exception" << endl;
}
catch (...) { // Catch Line-4
cout << "default" << endl;
}
cout << "end of program";
return 0;
}
```

Select one:

a.
```
Throw Line-1 - "caught logic_error"
Throw Line-2 - "default"
Throw Line-3 - "caught expB"
Throw Line-4 - "caught exception"
```

b.
```
Throw Line-1 - "caught exception"
Throw Line-2 - "default"
Throw Line-3 - "caught expA"
Throw Line-4 - "caught exception"
```

c. ✓
```
Throw Line-1 - "caught logic_error"
Throw Line-2 - "default"
Throw Line-3 - "caught expA"
Throw Line-4 - "caught exception"
```

d.
```
Throw Line-1 - "caught logic_error"
Throw Line-2 - "default"
Throw Line-3 - "caught expA"
Throw Line-4 - "caught logic_error"
```

Your answer is correct.
The correct answer is:
```
Throw Line-1 - "caught logic_error"
Throw Line-2 - "default"
Throw Line-3 - "caught expA"
Throw Line-4 - "caught exception"
```

Question 3
Complete
Marked out of 14.00
Flag question

Consider the following program which implements a range checking iterator for an array. Write the code for the three blanks such that the program prints the output provided below. [Marks 6+4+4=14]

Program:

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

template <typename T>
class Myarray {
    T * ptr;
    int size;
    public:
    Myarray<T>(T inp[], int n) : ptr(new T[n]),size(n) {
        for(int i=0; i<n; i++) ptr[i]=inp[i];
    }
    ~Myarray<T>() { delete[] ptr;}

    class Iterator {
        T *pointer;
        T *start;
        int size;

        public:
        Iterator(T *p = NULL) : pointer(p),start(p),size(-1) {}
        Iterator(T* p,T * st, int isize) : start(st),size(isize),pointer(p) {}
        T& operator*() const {
          ______BLANK 1____________
        }

          Iterator & operator=(const Iterator& other) {pointer=other.pointer;
          start=other.start; size=other.size; return *this;}
          friend bool operator== (const Iterator& a, const Iterator& b)
           { return a.pointer == b.pointer; };
          friend bool operator!= (const Iterator& a, const Iterator& b)
           { return a.pointer != b.pointer; };
          friend bool operator<= (const Iterator& a, const Iterator& b)
           { return a.pointer <= b.pointer; };
          friend bool operator< (const Iterator& a, const Iterator& b)
           { return a.pointer < b.pointer; };
          Iterator& operator++() { pointer++; return *this; }
          Iterator operator++(int) { Iterator tmp = *this; ++(*this); return tmp; }
      };

      Iterator begin() { _____BLANK 2_______ }
      Iterator end()   { _____BLANK 3_______ }
};

int main() {
    int temp[] = {1,2,3,4,5};
    Myarray<int> A(temp, 5);
    Myarray<int>::Iterator it;
    try{
        for(it=A.begin(); it<= A.end(); it++) cout << *it <<endl;
    }
    catch(out_of_range *e1) {
        cout<< "Out of range exception"<<endl;
    }
    return 0;
}
```

Output:

```
1
2
3
4
5
Out of range exception
```

Blank_1 :
return *pointer;

Blank_2 :
return Iterator(start);

Blank_3 :
return Iterator(start + size);

Question 4
Correct
Mark 4.00 out of 4.00
Flag question

What will be the outcome of compiling and running the following program: [Marks 4]

```
#include <iostream>
#include <vector>
#include <set>
#include <map>
#include <algorithm>
using namespace std;

int main() {
    int a[] = {1,2,3,2,3};
    vector<int> v;
    set<int> s;
    for(int i=0; i<5; i++) {
        v.insert(v.end(),a[i]);
        s.insert(a[i]);
    }

    vector<int>::iterator it1 = find(v.begin(), v.end(), 3);
    cout<<it1 - v.begin()<<", ";
    set<int>::iterator it2 = find(s.begin(), s.end(), 3);
    cout<< it2 - s.begin()<<endl;
    return 0;
}
```

Select one:
a. Compilation error ✓
b. `2, <Any of 2 or 4>`
c. `2, <Any number between 0 to 4>`
d. `2, 2`

Your answer is correct.
The correct answer is: Compilation error

Question 5
Correct
Mark 9.00 out of 9.00
Flag question

What will be the output of compiling and executing the following program? [Marks 9]

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

template <class T> class Sub;

template <typename T> class Obs {
    T msg;
    list<Sub<T>* > subs;
    public:
    Obs<T>(T a): msg(a) {}
    T getstate() {return msg; }
    void update() {
        typename list<Sub<T>* >::iterator it;
        for( it=subs.begin();it!=subs.end();++it) (*it)->update(msg);
    }
    void attach(Sub<T> *s) {subs.push_back(s);}
    void dettach(Sub<T> *s) {subs.remove(s);}
    friend void Sub<T>::notify();
};

template <class T> class Sub{
    Obs<T> *obs;
    T msg;
    int id;
    public:
    void setobs(Obs<T>* o) {obs=o;}
    void setid(int i) {id=i;}
    void notify() {
        if(id%2==0) {
            if(msg[0]<=msg[1]) obs->msg[0]++;
            else obs->msg[2]++;
        } else {
            if(msg[1]<=msg[0]) obs->msg[1]++;
            else obs->msg[2]++;
        }
    }
    void update(const T &imsg) { msg = imsg; }
};

int main() {
    vector<int> init_state(3);
    init_state[0]=init_state[1]=init_state[2]=1;
    Obs<vector<int>> o(init_state);
    Sub<vector<int>> s[10];

    for(int i=0;i<10;i++) { s[i].setid(i+1); o.attach(&s[i]); s[i].setobs(&o); }
    o.update();
    for(int i=0;i<10;i++) { s[i].notify(); o.update(); }

    vector<int> st=o.getstate();
    cout<<" "<<st[0]<<" "<<st[1]<<" "<<st[2]<<endl;
    return 0;
}
```

Select one:
a. `6 6 1` ✓
b. `6 2 5`
c. `1 1 11`
d. `6 5 2`

Your answer is correct.
The correct answer is: `6 6 1`

Question 6
Partially correct
Mark 1.00 out of 3.00
Flag question

Match the development activity with the appropriate SDLC Models. [LM, 0.5 * 6 = 3]

	Development Activity
(a)	I work with my friend as *driver / observer*
(b)	I design test cases and then code to make them pass
(c)	I do stand-up meeting with my team every morning
(d)	I build prototype and keep refining it in quick cycles
(e)	I use the most classical model for development
(f)	I repeat *planning, risk analysis, engineering, and evaluation*

	SDLC Models
(1)	TDD
(2)	RAD
(3)	SCRUM
(4)	Spiral
(5)	Waterfall
(6)	XP

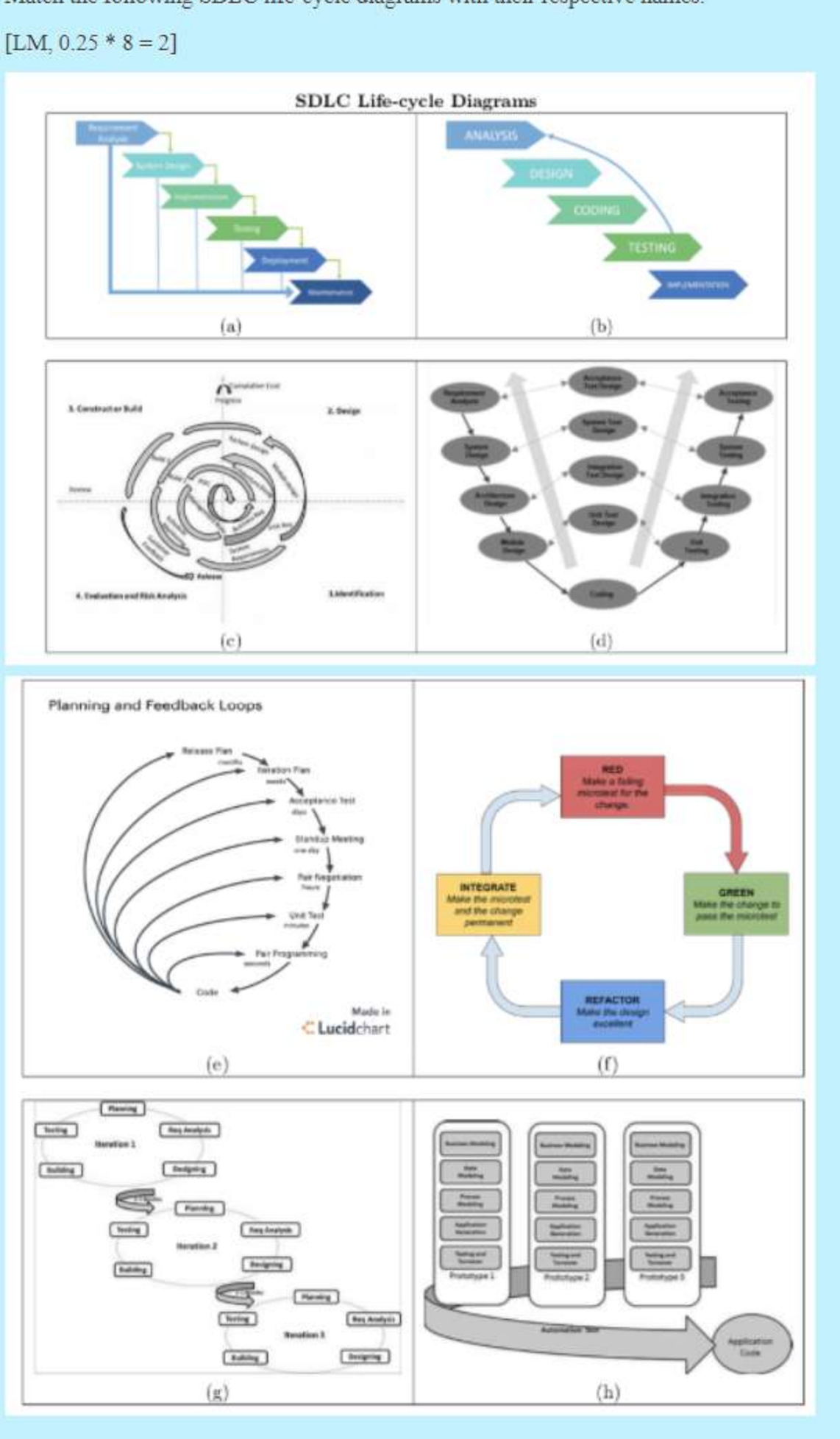
(b) (1) ✓
(c) (4) ✗
(a) (5) ✗
(f) (6) ✗
(e) (3) ✗
(d) (2) ✓

Your answer is partially correct.
You have correctly selected 2.
The correct answer is: (b) – (1), (c) – (3), (a) – (6), (f) – (4), (e) – (5), (d) – (2)

Question 7
Partially correct
Mark 1.50 out of 2.00
Flag question

Match the following SDLC life-cycle diagrams with their respective names.

[LM, 0.25 * 8 = 2]

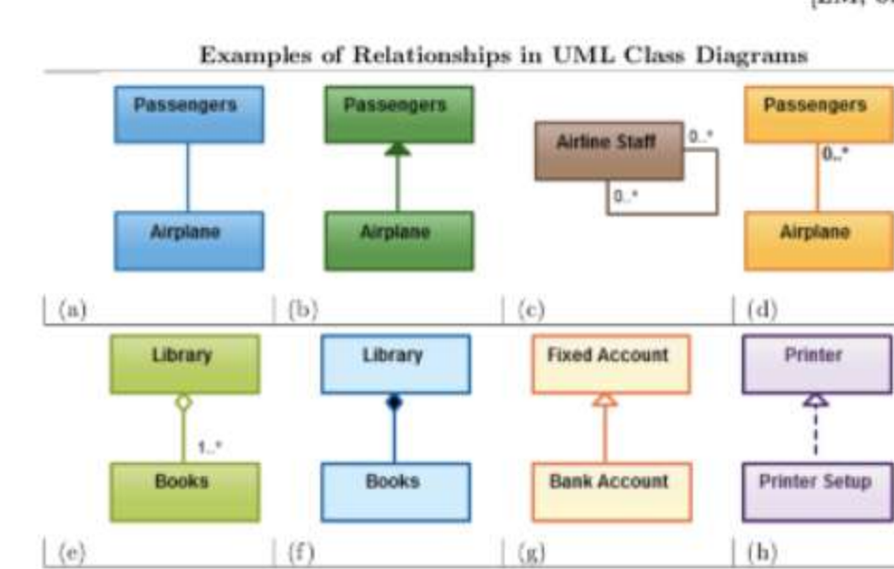


(e) Agile ✗
(b) Iterative ✓
(a) Waterfall ✓
(c) Spiral ✓
(g) XP ✗
(d) V ✓
(h) RAD ✓
(f) TDD ✓

Your answer is partially correct.
You have correctly selected 6.
The correct answer is: (e) – XP, (b) – Iterative, (a) – Waterfall, (c) – Spiral, (g) – Agile, (d) – V, (h) – RAD, (f) – TDD

Question 8
Not answered
Marked out of 2.00
Flag question

Match the illustrative examples below with UML Class Diagram Relationships. [LM, 0.25 * 8 = 2]



Class Diagram Relationships

(1)	Realization
(2)	Association
(3)	Inheritance / Generalization
(4)	Aggregation
(5)	Reflexive Association
(6)	Multiplicity
(7)	Composition
(8)	Directed Association

(e) Choose...
(g) Choose...
(h) Choose...
(c) Choose...
(f) Choose...
(d) Choose...
(b) Choose...
(a) Choose...

Your answer is incorrect.
The correct answer is: (e) – (4), (g) – (3), (h) – (1), (c) – (5), (f) – (7), (d) – (6), (b) – (8), (a) – (2)

Question 9
Not answered
Marked out of 3.00
Flag question

Consider the following *Quadratic Equation Solver* (QES) function `Solve` that takes 3 `double` parameters `a`, `b`, and `c` for solving equations of the form $ax^2 + bx + c = 0$. The solutions are passed back through output parameters `r1` and `r2`. The function returns a value designating the equivalence class of the root/s. [LM, 0.5 * 6 = 3]

The `Solve` function code is used in other questions too. So if you are getting this for the first time, you may study it well. Of course, the same will be provided in the other questions too where it is used.

```
00: unsigned int Solve(double a, double b, double c, double& r1, double& r2)
01: {
02:     unsigned int retVal = 0;
03:     if (0 == a) {
04:         if (0 == b) {
05:             if (0 == c) {
06:                 retVal = 5;
07:             } else {
08:                 retVal = 0;
09:             }
10:         } else { // Linear equation
11:             retVal = 1;
12:             r1 = -c/b;
13:         }
14:     } else {
15:         double disc = b*b - 4*a*c;
16:         if (0 == disc) {
17:             retVal = 2;
18:             r1 = r2 = -b/(2*a);
19:         } else {
20:             if (disc > 0) {
21:                 retVal = 3;
22:                 r1 = (-b + sqrt(disc))/(2*a);
23:                 r2 = (-b - sqrt(disc))/(2*a);
24:             } else {
25:                 retVal = 4;
26:                 r1 = -b/(2*a); r2 = sqrt(-disc))/(2*a);
27:             }
28:         }
29:     }
30:
31:     return retVal;
32: }
```

For checking the path coverage of `Solve`, a set of 6 test cases are designed below. Match the test cases with the paths it covers in the above code.

	Coefficients a	b	c
(a)	4	-12	9
(b)	4	0	9
(c)	6	-22	20
(d)	0	0	0
(e)	0	5	3
(f)	0	0	27

	Paths Covered
(1)	2-3-4-5-6-31
(2)	2-3-4-5-8-31
(3)	2-3-4-11-12-31
(4)	2-3-15-16-17-18-31
(5)	2-3-15-16-20-21-22-23-31
(6)	2-3-15-16-20-25-26-31

(a) Choose...
(c) Choose...
(b) Choose...
(d) Choose...
(e) Choose...
(f) Choose...

Your answer is incorrect.
The correct answer is: (a) – (4), (c) – (5), (b) – (6), (d) – (1), (e) – (2), (f) – (2)