X

 (https://swayam.gov.in)  (https://swayam.gov.in/nc_details/NPTEL)

vp2749@srmist.edu.in

Unit 10 - Week 8

Course outline

How does an NPTEL online course work?

Week 0

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

Week 8

Module 36 : Exceptions (Error Handling in C) : Part I (Lecture

Assignment 8

The due date for submitting this assignment has passed. **Due on 2020-11-11, 23:59 IST.**

Assignment submitted on 2020-11-11, 23:54 IST

1) ***2 points***

52) (unit?unit=89&lesson=90)
Module 37 : Exceptions (Error Handling in C) : Part II (Lecture 53) (unit?unit=89&lesson=91)
Module 38 : Template (Function Template) : Part I (Lecture 54) (unit?unit=89&lesson=92)
Module 39 : Template (Function Template) : Part II (Lecture 55) (unit?unit=89&lesson=93)
Module 40 : Closing Comments (Lecture 56) (unit?unit=89&lesson=94)
Lecture Materials (unit?unit=89&lesson=95)
Quiz : Assignment 8 (assessment?name=176)
W8_Programming-Qs1 (/noc20_cs57/progassignment?name=178)
W8_Programming-Qs2 (/noc20_cs57/progassignment?name=179)
W8_Programming-Qs3 (/noc20_cs57/progassignment?name=180)
W8_Programming-Qs4

Consider the program below.

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

void fun(int test) {
    try {
        test ? throw test : throw "zero ";
    }
    catch (int i) {                    //LINE-1
        cout << "Caught: " << i << " ";
    }
}
int main() {
    try{
        fun(1);
        fun(2);
        fun(0);
        fun(3);
    }
    catch (const char *str) {          //LINE-2
        cout << "CaughtString ";
    }
    return 0;
}
```

What will be the output?

a) `Caught: 1 Caught: 2 CaughtString Caught: 3`

b) `Caught: 1 Caught: 2 CaughtString`

c) `Caught: 1 Caught: 2 CaughtString zero Caught: 3`

d) `Caught: 1 Caught: 2`

Yes, the answer is correct.
Score: 2

Accepted Answers:

b) `Caught: 1 Caught: 2 CaughtString`

2) ***2 points***

(/noc20_cs57/progassignm name=181)

Feedback For Week 8 (unit? unit=89&lesson=96)

DOWNLOAD VIDEOS

Text Transcripts

Assignment Solution

Books

Live Interactive Session

Programming Test (11th Dec): Session-1 (10.00AM - 11.00AM)

Programming Test (11th Dec): Session-2 (8.00PM - 9.00PM)

Consider the program below.

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

namespace cust_error {
    class error { };
    class spec_error : public error { };
    class unknown_error : public error { };
    void f() { throw unknown_error(); }
};

int main() {
    try {
        cust_error::f();
    }
    catch (cust_error::spec_error&) {        // LINE-1
        cout << "specific error" << endl;
    }
    catch (cust_error::error&) {             // LINE-2
        cout << "error" << endl;
    }
    catch (cust_error::unknown_error&) {     // LINE-3
        cout << "unknown error" << endl;
    }
    catch (...) {                            // LINE-4
        cout << "default" << endl;
    }
    return 0;
}
```

What will be the output?

- a) specific error
- b) error
- c) unknown error
- d) default

Yes, the answer is correct.
Score: 2

Accepted Answers:

b) error

3) **2 points**

Consider the following program.

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

int main() {
    try {
        throw "s";
    }
    catch (int x) {
        cout << "Caught 1 " << x;
    }
    catch (char x) {
        cout << "Caught 2 " << x;
    }
    catch (string x) {
        cout << "Caught 3 " << x;
    }
    catch (...) {
        cout << "Default Exception";
    }
    return 0;
}
```

What will be the output?

- a) Caught 1
- b) Caught 2
- c) Caught 3
- d) Default Exception

Yes, the answer is correct.
Score: 2

Accepted Answers:

d) Default Exception

4) **2 points**

Consider the following program.

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

template<class T> T GetMax(T& a, T& b) {    // LINE-1
    return ((a>b) ? a : b);
}

int main() {
    int i = 5, j = 6, k;
    long l = 10, m = 5, n;

    k = GetMax<int>(i, j);
    n = GetMax<long>(l, m);
    cout << k << " ";
    cout << n;

    return 0;
}
```

Fill the blank at LINE-1, such that the output is:
6 10

- [x] a) `int GetMax (int a, int b)`
- [ ] b) `template <typename T> GetMax`
- [ ] c) `template <typename T> T GetMax(T a, T b)`
- [ ] d) `template <class T> T GetMax(T& a, T& b)`

No, the answer is incorrect.
Score: 0

Accepted Answers:

c) `template <typename T> T GetMax(T a, T b)`

d) `template <class T> T GetMax(T& a, T& b)`

5) **2 points**

Consider the code below.

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

template <typename T>
T sum(T x, T y) {
    return x + y;
}

int main() {
    cout << _____;    // LINE-1

    return 0;
}
```

What shall be the output/error when the blank space in LINE-1 is filled with the following:
(i) `sum(10, 20)`
(ii) `sum(3.14, 9.76)`
(iii) `sum(3.14, 9)`

- a) Error: For all the calls, type is not instantiated
- b) (i) 30, (ii) `12.9`, (iii) `12.14`
- c) (i) 20, (ii) 12, (iii) error: as no matching for `sum(double, int)`
- d) (i) 30, (ii) `12.9`, (iii) error: as no matching for `sum(double, int)`

Yes, the answer is correct.
Score: 2

Accepted Answers:

d) (i) 30, (ii) `12.9`, (iii) error: as no matching for `sum(double, int)`

6) **2 points**

Consider the following program.

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

____________________ // LINE-1
class List {
    T arr[N];
public:
    void setVal(int x, T value) {
        arr[x] = value;
    }
    T getVal(int x) {
        return arr[x];
    }
};

int main() {
    List<int, 5> myints;
    List <double, 5> mydoubles;

    myints.setVal(3, 10);
    mydoubles.setVal(1, 3.14);
    cout << myints.getVal(3) << " ";
    cout << mydoubles.getVal(1) << " ";

    return 0;
}
```

Fill in the blank at LINE-1 such that the output is:
10 3.14

- [ ] a) `template <class T>`
- [x] b) `template <typename T, int N = 0>`
- [ ] c) `template <class T, class N = 0>`
- [x] d) `template <class T, int N>`

Yes, the answer is correct.
Score: 2

Accepted Answers:

b) `template <typename T, int N = 0>`

d) `template <class T, int N>`

7) **2 points**

Consider the program below.

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

template <class T, int i>
void repeat(T val) {
    i = 5;
    for (int j = 0; j < i; j++)
        cout << val << " ";

    return;
}

int main() {
    repeat<int, 10>(10);

    return 0;
}
```

What will be the output / error?

- a) `10 10 10 10 10 10 10 10 10 10`
- b) `10 10 10 10 10`
- c) `10 10 10 10 10 0 0 0 0 0`
- d) `Compiler error: l-value required`

Yes, the answer is correct.
Score: 2

Accepted Answers:

d) `Compiler error: l-value required`

8) **2 points**

Consider the program below.

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

template<class T>
class Adder {
    T n1, n2;
public:
    Adder(T _n1, T _n2) :n1(_n1), n2(_n2) { }
    T Add();
};

_____          // LINE-1: Declare the Template
_____ {        // LINE-2: Fill with the correct Template signature
    return n1 + n2;
}

int main() {
    Adder<int> obj1(10, 20);
    Adder<double> obj2(3.14, 8.6);

    cout << obj1.Add() << " " << obj2.Add() << endl;

    return 0;
}
```

Fill in the blanks at LINE-1 and LINE-2 with appropriate options such that the output is:
30 11.74

a) LINE-1: template<class T>, LINE-2: T Adder<>::Add()

b) LINE-1: template<class T>, LINE-2: T Adder<T>::Add()

c) LINE-1: template<typename T>, LINE-2: T Adder::Add()

d) LINE-1: template<typename T>, LINE-2: T Adder<typename T>::Add()

Yes, the answer is correct.
Score: 2

Accepted Answers:

b) LINE-1: template<class T>, LINE-2: T Adder<T>::Add()

9) **2 points**

Consider the program below.

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

typedef struct complex_num {
    int r, i;
}COMPLEX;

template<class T>
T operator+(T& a, T& b) {
    return a + b;
}

----------                    // LINE-1
COMPLEX operator+(COMPLEX& a, COMPLEX& b) {
    COMPLEX c;
    c.r = a.r + b.r;
    c.i = a.i + b.i;
    return c;
}

int main() {
    int a = 10, b = 20;
    COMPLEX c1 = { 10, 20 };
    COMPLEX c2 = { 30, 40 };

   int c = a + b;
   cout << c << endl;

   COMPLEX c3 = c1 + c2;
   cout << c3.r << " , " << c3.i;

   return 0;
}
```

Fill in the blank at LINE-1 with appropriate option such that the output is:
30
40 , 60

- a) LINE-1: `template<>`

b) LINE-1: template<COMPLEX>

c) LINE-1: template

d) LINE-1: template<T>

Yes, the answer is correct.
Score: 2

Accepted Answers:

a) LINE-1: template<>