Computer Science & Engineering Department
I. I. T. Kharagpur

Principles of Programming Languages: CS40032

Elective

Assignment – 4: Lambda and Functors in C++ *Marks: 25*

---

1. Answer the following questions: **[13]**

   (a) What is the correct statement about lambda expression?

   i. The return type of lambda expression can be neglected in some cases

   ii. The return type of lambda expression must be specified in all cases

   iii. Lambda expression should be very large functions

   iv. Lambda expression is also available in C

   [1]

   (b) Write a code snippet containing a lambda expression in C++ which captures a double variable with value 12.3, takes a double variable as parameter, increments the captured variable and returns its summation with the parameter into an int variable (don't worry about loss of precision). **[2]**

   (c) Write the output for the following: **[1]**

```
#include<iostream>
using namespace std;
int main()
{
  int x = 5;
  auto check = []() -> bool
        {
    if(x == 0)
      return false;
    else
      return true;
  };
  cout<<check()<<endl;
  return 0;
}
a) 1
b) 0
c) Error
d) Segmentation fault
```

   (d) Write the output for the following: **[1]**

```
#include<iostream>
using namespace std;
int main()
{
  int a = 5;
  auto check = [=]()
        {
    a = 10;
  };
  check();
  cout<<"Value of a: "<<a<<endl;
  return 0;
}
a) Value of a: 5
b) Value of a: 10
c) Error
d) Segmentation fault
```

   (e) Write the output for the following: **[1]**

```
#include<iostream>
using namespace std;
int main()
```

```
{
  int a = 5;
  int b = 5;
  auto check = [&a]()
        {
    a = 10;
    b = 10;
  }
  check();
  cout<<"Value of a: "<<a<<endl;
  cout<<"Value of b: "<<b<<endl;
  return 0;
}
a) Value of a: 5
b) Value of a: 10
c) Error
d) Segmentation fault
```

(f) Write the output for the following: **[2]**

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

int main() {
  double c=3.3;
  auto func =  [c]() mutable -> int {
              ++c;
              cout << c << endl;
              return c;};
  func();
  auto cen = func();
  cout << cen << endl;
  auto l = [con = 7](int x) { return x+con; };
  cout << l(5);
  return 0;
}
```

(g) Write the output for the following: **[2]**

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

int main() {
  auto con= 7;
  auto l = [&](int x) { return x+con; };
  ++con;
  cout << l(5) << endl;
  auto m = [=](int x) { return x+con; };
  ++con;
  cout << l(5) <<          << m(5) << endl;
  return 0;
}
```

(h) Make apt changes to avoid compilation error without using 'mutable' or changing the logic in the following (Just write the changed lambda expression): **[1]**

```
#include <iostream>
using namespace std;
int main() {
  int c=3;
  auto func =  [=]()  -> int {++c; cout << c;
                      return 100.2;};
  func();
  return 0;
}
```

(i) Write the output: **[2]**

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

int main() {
  int c=3;
  auto func1 =  [=]() mutable {++c; cout << c;};
  func1();
  cout << c;
  auto func2 =  [&]() mutable {++c; cout << c;};
  func2();
  cout << c;
  return 0;
}
```

2. Write a C++ code to decide if a given number is between 0 and 10 using: **[4]**

   (a) Functors **[2]**

   (b) Lambda Expressions **[2]**

3. Write a C++ code to implement the 'Towers of Hanoi' problem using: **[8]**

   (a) Functors **[4]**

   (b) Lambda Expressions **[4]**

The Problem Statement:
Tower of Hanoi is a mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

1) Only one disk can be moved at a time.
2) Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
3) No disk may be placed on top of a smaller disk.

Credits: geeksforgeeks