

# Principles of Programming Languages: CS40032

## Assignment-III: $\lambda$ and *Functors in C++*

Arkajyoti Pal

February 10, 2019

1. (a) 

```
int age=25;
auto func=[age](int val){cout<<(age-val);};
```
- (b) 

```
double var=12.3;
auto func=[&var](double val)
{
    var+=1.0;
    return ((int)(var+val));
};
```
- (c) 

```
4.3
5.3
5
12
```
- (d) The code snippet, given as it is, would give a compilation error. Changing the line  

```
cout << l(5) <<    << m(5)<<endl ;
```

  
to  

```
cout << l(5) <<" " << m(5)<<endl ;
```

  
would solve the compilation error and give the following output:  

```
13
14 13
```
- (e) 

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

```

        func();
        return 0;
    }

```

(f) 4344

2. (a) `#include <bits/stdc++.h>`  
`using namespace std;`

```

class TowerOfHanoi{

```

```

    public:

```

```

        void operator()(int n, char from, char aux, char to)
        {

```

```

            if (n==1)
            {

```

```

                cout<<"\t\tMove disc 1 from " <<from<<" to " <<to<<"\n";
                return;
            }

```

```

        else
        {

```

```

            (*this)(n-1,from,to,aux);

```

```

            cout<<"\t\tMove disc " <<n<<" from " <<from<<" to " <<to<<

```

```

            (*this)(n-1,aux,from,to);
        }
    }

```

```

};

```

```

int main()
{

```

```

    TowerOfHanoi tower;

```

```

    tower(8, 'A', 'B', 'C');

```

```

    return 0;
}

```

- (b) `/* Solving Tower of Hanoi puzzle using Lambda Expressions */`

```

#include <iostream>

```

```

#include <functional>

```

```

using namespace std;

```

```

int main()
{

```

```

    std::function<void(int, char, char, char)> hanoi;

```

```

    hanoi=[&hanoi](int a, char from, char aux, char to) {

```

```

        if (a==1){

```

```

            cout<<"\t\tMove disc 1 from " <<from<<" to " <<to<<"\n";
        }
    }

```

```

        return ;
    }
    else {
        hanoi(a-1,from,to,aux);
        cout<<"\t\tMove disc \t" <<a<<" from \t" <<from<<" to \t" <<to<<"\n";
        hanoi(a-1,aux,from,to);
    }
};
cout<<" hanoi (8): \t" <<endl;
hanoi(8, 'A', 'B', 'C');
}

```

3. (a)

$$(\lambda x.x^2(\lambda x.(x+1)2))$$

*Applicative order :*

$$\Rightarrow (\lambda x.x^2(\lambda \underline{x}.(x+1)\underline{2}))$$

$$\Rightarrow (\lambda \underline{x}.x^2\underline{3})$$

$$\Rightarrow 9$$

(b)

*Normal order :*

$$\Rightarrow (\lambda \underline{x}.x^2(\lambda x.\underline{(x+1)2}))$$

$$\Rightarrow (\lambda x.(x+1)2)^2$$

$$\Rightarrow (\lambda \underline{x}.\underline{(x+1)2}) * (\lambda \underline{x}.\underline{(x+1)2})$$

$$\Rightarrow 3 * 3$$

$$\Rightarrow 9$$