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

Software Engineering: CS20006/CS20202

Assignment – 1: Better C & Guidelines *Marks: TBA*

Assign Date: *12th January, 2022* Submit Date: *23:55, 23rd January, 2022*

Instructions: Please solve the questions using pen and paper and scan the images. Every image should contain your roll number and name.

1. Is the following program correct?

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

#define pi (22/7)
#define area(r) (pi * r * r)

int main()
{
     int radii[] = {3,4,5};
     double totalarea=0;
     for(int i=0; i<3; i++) totalarea += area(radii[i]);
     cout << totalarea << endl;
     return 0;
}
```

   If not, identify the bug and fix it. **[1 + 1 = 2]**

   Can the same bug occur if `area()` was defined as an inline function ? Why ? **[1]**

2. Identify and fix the bug in the following program. **[1]**

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

const int inc(const int a) { return a++; }

int main() {
    cout << inc(5) <<endl;
}
```

   What is a situation in which returning a const value is good idea? **[1]**

3. Consider the following program:

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

int rem(int n, int r) {
    return n % r;
}

int main() {
    const int n = 15, r = 0; //line 1

//  int n , r; // Line 2
//  cin >> n >> r; //Line 3
```

```
    if (r == 0 || rem(n, r))
        cout << "True" << endl;
    else
        cout << "False" << endl;

    if (rem(n, r) || r == 0)
        cout << "True" << endl;
    else
        cout << "False" << endl;

    return 0;
}
```

While using the GCC c++ compiler, the output is as follows:

Build Type	Output
(Un-optimized) `g++`	`True` `Floating Point Exception`
(Optimized) `g++ -O1`	`True` `True`

If we comment `Line 1` and un-comment `Line 2` and `Line 3`, the output changes to the following: while we input `15` for `n` and `0` for `r` (as was initialized in `Line 1`:

Build Type	Output
(Un-optimized) `g++`	`True` `Un-handled Floating Point Exception`
(Optimized) `g++ -O1`	`True` `Un-handled Floating Point Exception`

Explain the behavior in both cases, especially justifying the difference due to changing `Line 1` to `Line 2` & `Line 3` and providing the same input. **[2 + 2 = 4]**

Write an appropriate guideline to avoid such bugs and improve the quality of the code. **[1]**

4. Is there a bug in the following program ? If yes, identify the bug, otherwise write the output and explain it.

[2]

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

int f1(int a,float b) {
    cout <<a << " " <<b <<"\n";
    return 0;
}

int f1(int, float b=30);
int f1(int a=10, float );

int main() {
    f1(10,20);
    f1(10.4);
    return 0;
}
```

5. Define an appropriate function `min` such that in following program snippet, the minimum of the variables `c` and `d` is assigned the value 1.

[2]

```
int main() {
    int c,d;
    cin >> c >>d;
    cout <<c <<" "<< d<<"\n";
    min(c,d) = 1;
    cout <<c <<" "<< d<<"\n";
    return 0;
}
```

6. Which of the following sets of function definitions are legal ? Mention reasons for each case.

[5 × 2 = 10]

(a)
```
int fun(int *ptr, int n) {...};
int fun(int ptr[], int n) {...};
```

(b)
```
int fun(int **ptr, int n) {...};
int fun(int ptr[][], int n) {...};
```

(c)
```
int fun(int x, int y);
static int fun(int x, int y);
```

(d)
```
void fun(int x, int y) {cout<<"f1\n";}
void fun(int &x, int y)  {cout<<"f2\n";}
```

(e)
```
void fun(int *x, int y) {cout<<"f1\n";}
void fun(int &x, int y)  {cout<<"f2\n";}
```

7. Which of the following operator overloads are valid ? Give reasons.

[3 × 2 = 6]

- ```
  typedef int Int;
  Int operator+(Int a,Int b) { Int c; c = a + b + 5; return c;}
  ```
- ```
  typedef struct _int { int v; } MyInt;
  MyInt operator+(MyInt a,MyInt b) {MyInt c; c.v = a.v+b.v+5; return c;}
  ```
- ```
  struct MyInt { int v; };
  int operator+(MyInt a,MyInt b) { return a.v+b.v+5; }
  ```

8. What will be the output for the following code ? Explain.

[2 + 2 = 4]

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

struct MyInt { int v; };
MyInt operator++ (MyInt a, int ) { a.v++; return a;}

int main() {
    MyInt a;
    a.v=5;
    a++;
    cout<<a.v;
}
```

Modify the above program to achieve the desired output.
```