CS 101 (Autumn 2025)
Midsem exam

16 Sep 2025 - 08:30 AM to 10:30 AM

Instructor: Rohit Gurjar

Roll Number	SAMPLE
Division and Group	
Name	

Q.No.	**Marks** / **max**	**Graded By**	**Verified By**	**Re-grading** (if requested)
1	/ 4			
2	/ 3			
3	/ 6			
4	/ 5			
5	/ 3			
6	/ 5			
7	/ 4			
8	/ 4			
9	/ 6			
TOTAL	/ 40			

Please read the following instructions carefully before you start.

- Write your roll number, name, and group number in the space provided. A paper without a roll number and name will NOT be graded.
- Write your answers neatly with a blue/black pen on this question paper itself in the space provided for each question. At the end, you must submit this paper to the invigilator.
- Rough pages will NOT be provided. You can use the space given for rough work in the question paper.
- Please note that your answers should NOT include any programming concept that hasn't been covered in the class so far. If such answers are found, they shall NOT be graded.
- No clarifications will be provided on any questions. When in doubt, make suitable assumptions, state them clearly, and proceed to solve the problem. If your answer depends on any assumption you have made, the assumption must be explicitly stated in your paper.
- In some questions, you are provided code snippets. Unless specified as being buggy, assume that the code snippet is enclosed suitably within the main, correct header files are included, etc., and therefore, the code compiles.
- All the best!

Translation

- शुरू करने से पहले कृपया नीचे दिए गए निर्देशों को ध्यान से पढ़ें।
- दिए गए स्थान पर अपना रोल नंबर, नाम और ग्रुप नंबर लिखें। बिना रोल नंबर और नाम के पेपर को grade नहीं दिया जाएगा।
- अपने उत्तर इस प्रश्नपत्र पर ही प्रश्न के साथ दिए गए स्थान पर नीले/काले पेन से साफ-सुथरे लिखें। अंत में आपको यह पेपर निरीक्षक के पास जमा करना होगा।
- रफ पेज उपलब्ध नहीं कराए जाएंगे. रफ कार्य के लिए प्रश्नपत्र में ही जगह दी गई है.
- ध्यान रहे कि आपके उत्तरों में ऐसे किसी भी programming concept का इस्तेमाल नहीं होना चाहिए जिसे अब तक कक्षा में पढ़ाया नहीं गया है। यदि ऐसे उत्तर पाए जाते हैं, तो उन्हें grade नहीं किया जाएगा।
- किसी भी सवाल पर कोई स्पष्टीकरण नहीं दिया जाएगा. यदि सवाल हल करने के लिए आपको किसी assumption की जरूरत लगती है तो, तो स्पष्ट रूप से अपना assumption लिखें और उसके आधार पर सवाल हल करें.
- कुछ प्रश्नों में आपको code का एक हिस्सा दिया गया है। ये मानते हुए चलें कि code का ये हिस्सा main program के अंदर लिखा है, और सही header files include की गयी हैं, इत्यादि, और इसलिए, code सही से compile होता है।
- शुभकामनाएँ!

Question 1 [2+2 marks]

```
bool g(bool x, bool y){
    if(x)
        return y;
    else
        return true;
}
```

(a) Consider the above defined function `g`. Write a Boolean expression in Boolean variables `p` and `q` which is equivalent to `g(p,q)`. That is, for all possible Boolean values of `p` and `q`, the expression should give the same value as `g(p,q)`. Your expression should only use variables `p`, `q` and some operators from `&&`, `||`, and `!`. Ternary operator is not allowed.

(a) ऊपर define किया गए function `g` पर विचार कीजिए. एक Boolean expression लिखिए जिसमें Boolean variables `p` और `q` हों, जो `g(p,q)` के समकक्ष हो. मतलब कि, `p` और `q` की सभी संभावित Boolean values के लिए वह expression वही values दे जो `g(p,q)` देता है. आपके expression में केवल variables `p`, `q` और `&&`, `||`, तथा `!` में से कुछ operators का उपयोग होना चाहिए. Ternary operator की उपयोग नहीं किया जा सकता.

(b) Suppose we have defined a function `area` which computes the area of a regular polygon. It has two parameters; first parameter is of `int` type denoting number of sides and second parameter is of `double` type denoting the side length. It returns the area (`double` type) of the polygon.

Suppose we have defined the function `area` in a file named `polygon.cpp`. This function is being called in another file `main.cpp`, which is given on the right-hand side. We will compile `main.cpp` and `polygon.cpp` together.

```
// main.cpp
#include<simplecpp>
#include "polygon.h"
main_program{
    cout << area(10, 2.5) << endl;
}
```

What should be written in the header file `polygon.h` for this to work correctly?

ऐसे में `polygon.h` नामक header file में क्या लिखा जाना चाहिए ताकि यह प्रोग्राम सही से काम करे?

Your answer:

(a) `!p || q`
OR
`!(p && !q)`
OR
`!p || (p && q)`
OR
any other equivalent expression

(b) `double area(int, double);`
or
`double area(int nSides, double length);`

here the variable names can be arbitrary.

Marks:
2 + 2 marks.
(a) Acceptable if answer is in terms of x,y instead of p,q
1 mark if written if(!p || q) return true else return false
(b) No deduction if missed semicolon
No deduction if header guards added
-1 if function name is incorrect
-1 if one of the types is wrong
0 marks if the complete function definition is written -- that will not compile because definition is already there in polygon.cpp.
0 marks if <> are used instead of ()

Question 2 [1+1+1 marks]

```
#include<simplecpp>
main_program{
  int n;
  cin >> n;
  int p;
  for(p=1; ___(a)___; ___(b)___){}
  cout << p << endl;
}
```

The above program takes a positive integer n and prints the smallest power of 2 that is greater than or equal to n. For example, for $n = 6$ it will print 8 and for $n = 16$, it will print 16.

Fill in the blanks so that the program works correctly.

ऊपर दिया गया program एक positive integer n लेता है और ऐसी सबसे छोटी संख्या print करता है जो 2 की घात (power) हो और n से बड़ी या उसके बराबर हो। उदाहरण के लिए, यदि $n = 6$ हो तो यह 8 print करेगा और यदि $n = 16$ हो तो यह 16 print करेगा।

खाली स्थान भरिए ताकि program सही से काम करे.

```
#include<simplecpp>
main_program{
  int sum = 0;
  for(int i = 0; i < 10; i++){
    if(i%2 != 0) continue;
    sum = sum + i;
    if(i == 7) break;
  }
  cout << sum << endl;
}
```

(c) What will the above program print?

(c) ऊपर दिया गया program क्या print करेगा?

Your answer:

(a) `p < n`

(b) `p = p * 2` OR `p *= 2`

(c) 20

Marks:
1 mark for each.
(a) -0.5 if written p<=n or p<n+1
(b) -0.5 if written 2p instead of 2*p

Rough work

Question 3 [6 marks]

```
1  #include<simplecpp>
2  bool isPositive(int n){
3    if(n>0) {bool b = true;}
4    else {bool b = false;}
5    return b;
6  }
7  void print(int n){
8    if(n==1) {
9      cout << 1 << endl;
10   }
11   print(n-1);
12   cout << n << endl;
13 }
14 int sum(int m, int n){
15   int sum;
16   for(int i=m; i<=n; i++){
17     sum = sum+i;
18   }
19   return sum;
20 }
21 float limit(float x){
22   float s;
23   if(x == 0) s = 1;
24   if(x < 0) s = sin(x)/x;
25   else s = sin(x)/x;
26   return s;
27 }
28 main_program{
29   int n;
30   cin >> n;
31   if(isPositive(n)) print(n);
32   cout << sum(3) << endl;
33   cout << limit(0) << endl;
34 }
```

The above program contains multiple mistakes, each of which can cause a compilation error, or a segmentation fault (due to infinite recursion), or the program to run indefinitely, or show some unexpected behavior. Point out any 4 mistakes with line numbers, and for each mistake say what it will cause. You will get 1.5 marks for each mistake.

ऊपर दिए गए program में कई गलतियाँ हैं. इनमें से हर एक गलती compilation error करा सकती है, या segmentation fault (infinite recursion की वजह से), या उसके कारण program अनंत समय तक चल सकता है, या फिर कुछ unexpected behavior दिखा सकती है. किसी भी 4 गलतियों को line numbers के साथ बताइए, और प्रत्येक गलती के लिए लिखिए कि उसका परिणाम क्या होगा. प्रत्येक गलती के लिए आपको 1.5 अंक मिलेंगे.

Your answer:

(i) Line 5: no variable b in this scope. Compilation error.

(ii) Lines 8-10: return is missing. Thus, when n=1, print(0) will be called leading to infinite recursion.

(iii) Line 15: sum is not initialized. Garbage value.

(iv) Line 25: if x=0, line 25 will get executed assigning 0/0. NaN.

(v) Line 32: function sum requires 2 arguments, but only one is given. Compilation error.

Marks:
1.5 mark for each mistake.
0.5 marks for line number,
1 mark for describing the mistake.
Line numbers need not be exactly as given above.
Any four mistakes can be mentioned.

This is not a mistake --
Line 15: variable sum inside function sum will cause compilation error.

Some students have given corrected lines instead of pointing the mistake. Should get marks.

Rough work

Question 4 [2+3 marks]

```
#include<simplecpp>
void f(int n){
  if (n==0)
    return;
  else{
    repeat(4){
      forward(50);
      f(n-1);
      left(90);
    }
  }
}

main_program{
  turtleSim();
  int n;
  cin >> n;
  f(n);
  hide();  getClick();
}
```

The above program uses a function to make a turtleSim drawing. Make the drawing for (a) n=1 and (b) n=2.

You will get marks only if the drawing is completely correct.

ऊपर दिया गया program एक function का उपयोग करके turtleSim drawing बनाता है. (a) $n = 1$ और (b) $n = 2$ के लिए drawing बनाइए.

आपको अंक तभी मिलेंगे जब drawing पूरी तरह सही होगी.

Your answer:

(a)

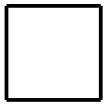

Figure 1: Answer (a)

(b)

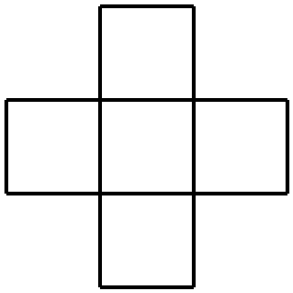

Figure 2: Answer (b)

Marks: (a) 2 marks (b) 3 marks.

Marks only if drawing is completely correct.
It is acceptable if edges are not perfect straight lines or if edge lengths are not exactly equal or if angles are not perfect 90 degrees.
Showing side lengths or angles not expected. No deduction if it is shown.
No deduction if turtle (triangle) position is shown.

Rough work

Question 5 [1+2 marks]

```
1 int f(int n, int f1=1, int f0=0){
2   if(n==0)
3     return f0;
4   if(n==1)
5     return f1;
6   else
7     return f(n-1)+f(n-2);
8 }
```

Recall that Fibonacci function is defined with the following recurrence relation

याद करें कि Fibonacci function को निम्नलिखित recurrence relation से define किया जाता है:

$$f(n) = f(n-1) + f(n-2).$$

Usually, the initial values are taken as $f(0) = 0$ and $f(1) = 1$. In some applications, you may encounter a function with the same recurrence relation, but with different initial values. For example, with initial values $f(0) = 2$ and $f(1) = 5$, we will get $f(2) = 7$, $f(3) = 12$ and so on.

आम तौर पर Fibonacci function की शुरुआती values ली जाती हैं: $f(0) = 0$ और $f(1) = 1$. कुछ applications में आपको वही recurrence relation मिलेगा, लेकिन शुरुआती values अलग हो सकती हैं. उदाहरण के लिए, यदि शुरुआती values $f(0) = 2$ और $f(1) = 5$ ली जाएँ, तो हमें मिलेगा $f(2) = 7$, $f(3) = 12$ और ऐसे ही आगे की values.

Above we want to define a Fibonacci function with three parameters, first is an integer n, and the other two parameters are initial values. The idea is that if the values for `f0` and `f1` are not provided, then the default values should be taken as 0 and 1, respectively. Below are some expected return values of function calls.

ऊपर हम एक Fibonacci function तीन parameters के साथ define करना चाहते हैं. पहला parameter एक integer n होगा, और बाकी दो parameters शुरुआती values होंगी. हम चाहते हैं कि अगर `f0` और `f1` की values नहीं दी जाती हैं, तो default values 0 और 1 ली जाएँ. नीचे कुछ function calls की वांछित return values दी गई हैं.

Function call	Expected return value
`f(3)`	2
`f(3,5,2)`	12

But the return values are not as expected.

(a) What is the return value of function call `f(3,5,2)`?

(b) The desired behaviour can be obtained by changing one line in the code. Write the line number and the corrected line.

लेकिन return values उम्मीद के अनुसार नहीं हैं.

(a) function call `f(3,5,2)` की return value क्या होगी?

(b) इच्छित व्यवहार (desired behaviour) केवल एक पंक्ति बदलकर प्राप्त किया जा सकता है. पंक्ति संख्या और सुधरी हुई पंक्ति लिखिए.

Your answer:

(a) 2

(b) Line No. 7
Corrected line:
return f(n-1,f1,f0)+
f(n-2,f1,f0);

Marks:
1+2

no deduction if "return"
missed.

0.5 marks if line number 7
is written but, correction is
wrong

No deduction if line number
missed, but the answer is
correct.

-0.5 if order of f1, f0
swapped.

only 0.5 marks if written
f(n-1,5,2)+ f(n-2,5,2)

Rough work

Question 6 [5 marks]

```
1  #include<simplecpp>
2  bool palindrome(int n){
3    // Pre-condition: n >= 0.
4    if(n == 0) return true;
5    char first, last;
6    if(n == 1){
7      ___(a)___ ;
8      return ___(b)___ ;
9    }
10   cin >> first;
11   bool middle = palindrome(___(c)___);
12   cin >> last;
13   if(first == last)
14     return ___(d)___;
15   else
16     return ___(e)___;
17 }
18
19 main_program{
20   int n;
21   cin >> n;
22   cout << palindrome(n) << endl;
23 }
```

A sequence of characters is called palindrome if it reads the same backwards as forwards. For example, madam, abacaba, xyyx, pp, q. The above program has a function `palindrome(int n)` whose task is to read a sequence of `n` characters and return true if it is a palindrome, and false otherwise. Its definition uses recursion. On the right-hand side there are some examples of input/output for the above program (true is printed as 1 and false as 0).

अक्षरों की एक sequence को palindrome कहते हैं यदि वह आगे से और पीछे से पढ़ने पर एक ही हो. उदाहरण के लिए: madam, abacaba, xyyx, pp, q. ऊपर दिए गए program में एक function `palindrome(int n)` है. इसका काम है `n` अक्षरों की sequence पढ़ना और अगर वह palindrome हो तो true return करना, अन्यथा false return करना. इसकी definition में recursion का उपयोग किया गया है. दाईं ओर कुछ input/output के उदाहरण दिए गए हैं (true को 1 और false को 0 print किया जाता है).

Input	Output
4 abba	1
5 abcba	1
5 abcbc	0
4 abab	0

Fill in the blanks so that the program works correctly. Each blank should be a single statement/expression.

खाली स्थान भरिए ताकि program सही तरीके से काम करे. प्रत्येक खाली स्थान में केवल एक ही statement या expression होना चाहिए.

Your answer:

(a) `cin >> first` OR `cin >> last`

(b) `true(or any nonzero number)`
OR
`palindrome(n-1)`
OR
`palindrome(0)`

(c) `n - 2`

(d) `middle`

(e) `false` OR `0`

Marks:
1 mark for each.
(e) n=0 acceptable.

Rough work

Question 7 [1+1+2 marks]

```
1  #include<simplecpp>
2  void sort(int a, int &b){
3    if(a>b){
4      int temp = b;
5      b = a;
6      a = temp;
7    }
8  }
9  void sort(int &a, int &b, int &c){
10   sort(a,b);
11   sort(b,c);
12   sort(a,c);
13 }
14 main_program{
15   int a, b, c;
16   cin >> a >> b >> c;
17   sort(a,b,c);
18   cout << a << " " << b << " " << c << endl;
19 }
```

The purpose of the above program is to take 3 integers and print them in increasing order. However it is not working correctly.

(i) What will it print for the following input:
3 5 1

The program can be corrected by correcting line 2 and line 12.

(ii) What should be line 2?

(iii) What should be line 12? Write only one statement.

ऊपर दिए गए program का उद्देश्य 3 integers लेना और उन्हें बढ़ते क्रम (increasing order) में print करना है. लेकिन यह सही से काम नहीं कर रहा है.

(i) निम्नलिखित input के लिए यह क्या print करेगा:
3 5 1

इस program को line 2 और line 12 को ठीक करके सुधारा जा सकता है.

(ii) line 2 क्या होनी चाहिए?

(iii) line 12 क्या होनी चाहिए? केवल एक statement लिखिए.

Your answer:

(i) `3 5 5`

(ii) `void sort(int &a, int &b){`
OR
`void sort(int& a, int& b){`

(iii) `sort(a,b);`

Marks:
1+1+2 marks.

In (b) the arguments should be written correctly with call by reference. Other mistakes should be ignored.

Rough work

Question 8 [2+2 marks]

```
1  #include<simplecpp>
2
3  double integrate(function<double(double)> f,
4  double x1, double x0=0, double dx=0.0001){
5    double x = x0;
6    double sum = 0;
7    while(x < x1){
8      sum = sum + f(x)*dx;
9      x = x + dx;
10   }
11   return sum;
12 }
13
14 ____(a)____
15
16 main_program{
17   cout << ____(b)____ << endl;
18 }
```

Above we have defined a function `integrate` that approximately computes the integration of a given function f from `x0` to `x1`. (You can think of line 3 and line 4 being in a single line).

We want to call this function in the `main_program` and compute the integration of the function $g(x) = x^2$ from -3 to 3.

(a) Write the definition of a function named `xSquare` computing the function $g(x) = x^2$, which can be passed to function `integrate` as the first argument. The definition of `xSquare` will be put before the `main_program`.

(b) Fill in the blank in line 17 with a function call to `integrate` with `xSquare` as one of the arguments, that will give us an approximate value of

$$\int_{-3}^{3} x^2 \, dx.$$

Your answer:

(a) `double xSquare(double x) { return x*x; }`
OR
```
double xSquare(double x)
{double s = x*x; return s; }
```
OR
```
auto xSquare = [](double
x){return x*x;};
```

(b) `integrate(xSquare, 3, -3)`
OR
```
integrate(xSquare, 3, -3,
0.0001)
```
OR
`2*integrate(xSquare,3)`
OR
`2*integrate(xSquare,3,0)`
OR
`2*integrate(xSquare,3,0,0.0001)`

xSquare can be replaced with any other name, but it should be same in both (a) and (b).

Marks: 2+2 marks

part(a): -1 if any of the two types is wrong.
-1 if return statement is missing/wrong.

part(b): -0.5 if -3, 3 are in the opposite order.
Fourth argument is optional and can be given any small number.
-0.5 if the fourth argument is larger than 0.1.
0 marks if the first argument is not correct e.g., xSquare(x)

Rough work

Question 9 [2+4 marks]

```
1  #include<simplecpp>
2  int charToInt(char x, char y){
3    return ____(a)____;
4  }
5  main_program{
6    int n;
7    cin >> n; // no of digits
8    int rem = 0;
9    int sqroot = 0;
10   for(int j=0; j<n/2; j++){
11     char x0, x1;
12     cin >> x0 >> x1;
13     rem = rem*100+ charToInt(x0, x1);
14     int i = 9;
15     int g;
16     while(true){
17       g = ____(b)_____;
18       if (g <= rem) break;
19       i--;
20     }
21     rem = rem - g;
22     sqroot = sqroot * 10 + i;
23   }
24   cout << sqroot << endl;
25 }
```

The identity $(a+b)^2 = a^2+2ab+b^2$ may be helpful.

The above program first reads a number `n`, which is always assumed to be even. Then it reads an `n`-digit integer N digit by digit (as characters). The program comptues the square root of N one digit at a time, and finally prints it. (If N is not a perfect square, then it prints the integer part of the square root). Below are some example input output pairs.

Input	Output
2 49	7
4 1000	31
4 0144	12

The function `charToInt` takes two character arguments (which are assumed to be digits) and returns the two digit number made by them. For example, function call `charToInt('2', '5')` returns 25.

Fill in the blanks in line 3 and line 17 appropriately.

Hint: it may be helpful to do a dry run (execute the program manually) on input `4 2025`

Your answer:

(a) `(x-'0') * 10 + (y-'0')`
OR
`(x-48) * 10 + (y-48)`
OR
`x*10+y - 528`
OR
`(int(x)-int('0')) * 10 + (int(y)-int('0'))`

(b) `(20*sqroot + i) * i`
OR
`20*sqroot*i + i*i`

the two expressions can be written in different ways.

Marks:
2+4 marks.

(a) -1 if x and y swapped.
-0.5 if some bracket is missing
0 marks if written x or x - 0 instead of x - '0'
-0.5 if x0,x1 instead of x,y
(b) -1 if written 2 or 10 instead of 20.
-0.5 if i^2 instead of i*i, or any other * missing.
-1 if sq written instead of sqroot

Rough work