

BTH001

1. Assume the class Person containing this declaration

```
Person(string name = "?", int birthyear = 0, string email = "?");
```

In that case, which of the following is/are correct?

- Person p;
- Person p(2003, "andrew@bth.se");
- Person p = new Person("Andrew");
- Person p("Mrs x", 2001, "mrsx@bth.se");

(2p)

2. Assume you don't declare any constructor in a class. Is it then possible to create objects of that class?

(1p)

3. Assume the class Ball containing a default constructor and the member function void bounce().

Also assume that the following C++ code exists in the main function:

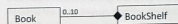
```
Ball* ballOne = new Ball();
Ball ballTwo;
```

Which of the following is/are correct?

- ballOne->bounce();
- ballTwo->bounce();
- (*ballOne).bounce();

(3p)

4. Assume the class diagram



Decide suitable member variable(s) for implementing the relationship between Book and BookShelf. Remember to describe in which class the member variable(s) you decided on exist. Also describe in what way the member variables will be used.

(4p)

5. Draw a class diagram that includes inheritance, that correspond to the description below. Use + and - for all members.

A player has the properties lives and number of collected items. An enemy has the properties lives, type and damage factor. For a player it must be possible to increase the amount of collected items. For an enemy it must be possible to get the damage factor. For both player and enemy, it must be possible to change the number of lives and also to get a string containing the content of all properties. The class diagram should not contain anything else than what is described above.

(5p)

6. Assume the classes Boat and SailBoat and that Sailboat inherits from Boat. Also assume that Boat has the constructor Boat(string name) and SailBoat has the constructor SailBoat(string name, double sailArea). Furthermore, assume that SailBoat only has the member variable double sailArea. Write C++ code to

- implement the relationship Boat inherits from SailBoat.
- define the constructor in class SailBoat. It must make a call of the base class constructor.

(3p)

(3p)

7. What is overriding?

(3p)

8. Assume the class Ball containing a default constructor. After the code below has executed

```
Ball bOne;
Ball* bTwo=new Ball();
Ball* bThree = new Ball[3];
Ball bFour[2];
Ball* bFive[5];
```

- how many Ball objects are there in the Stack (the statically part of the memory)?
- how many Ball objects are there in the Heap (the dynamically part of the memory)?

(3p)

(3p)

9. Assume that

class Base contains the following declarations of member functions

```
void funcOne();
virtual void funcTwo();
virtual void funcThree() =0;
```

class Sub1 which inherits from Base contains the following declarations of member functions

```
void funcOne();
void funcThree();
```

class Sub2 which inherits from Base contains the following declarations of member functions

```
void funcTwo();
void funcThree();
```

Also assume the following

```
Base* s1 = new Sub1(...);
Sub1* s2 = new Sub1(...);
Base* s3 = new Sub2(...);
Sub2* s4 = new Sub2(...);
```

For each call of member functions below specify the class (Base, Sub1 or Sub2) from which the function is executed.

- s1->funcOne();
- s4->funcOne();
- s3->funcTwo();
- s1->funcThree();

(8p)

10. What is required to make a class abstract? Give the syntax in C++.

(4p)

11. Assume the class Ball and that the class contains default-constructor, copy-constructor, assignment-operator and destructor. Give examples in C++ code when the

- default-constructor
- copy-constructor
- assignment-operator
- destructor

(1p)

(3p)

(2p)

(1p)

is executed.

12.

- Describe shallow copying and deep copying.
- Assume the class Person and the declaration `Person* person = new Person(...);`. Write C++ code that is an example of shallow copying.
- Assume the class Car and the declaration `Car* cars = new Car[5];`. Write C++ code that is an example of deep copying.

(4p)

(2p)

(3p)

13. Assume the class diagram below.



Also assume that the Ball is a class in which no member variable is declared as a pointer. Furthermore, assume that the member variables in BallContainer are

```
int capacity;
int nrOfBalls;
Ball* *balls;
```

and that there are member functions for adding and removing balls in the BallContainer class.

Write the definition (head and body) of

- the default constructor in BallContainer. The capacity shall be set to 5.
- the destructor in BallContainer.
- the copy-constructor in BallContainer.

(4p)

(6p)

(6p)

Deep copying is required.

14. Assume the classes Boat and SailBoat and that Sailboat inherits from Boat. Also assume that the function `double getSailArea()` is a function in class SailBoat (not in class Boat). Furthermore, assume the following declaration

```
Boat* boat = new SailBoat("Zara", 7.8);
```

- Write the C++ code needed to call the member function `getSailArea()` for the SailBoat object pointed at from the variable `boat`.
- Write the C++ code needed to create a copy of the SailBoat object above. The variable `SailBoat *sailboat` shall point to the copy of the SailBoat object. Observe - there is no member function available to get the name from the object.

(5p)

(8p)

15. Describe dynamic (late) binding and explain what is required to achieve dynamic binding.

(6p)

16. Assume the base class Boat and the sub class SailBoat and that none of the classes has member variables that are pointers. Also assume the following C++ code

```
Boat* boat = new SailBoat(...);
// ...
delete boat;
```

What is needed in the inheritance hierarchy to avoid memory leaks when the code is executed?

(3p)

17. Assume the function declaration

```
void remove(string arr[], int &nOf, string toRemove);
```

Modify the declaration to become a template function.

(4p)